



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PELELANGAN ONLINE (E-AUCTION) BERBASIS PERANGKAT BERGERAK ANDROID

ANDRE SETIAWAN
NRP. 5113 100 013

Dosen Pembimbing 1
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing 2
Rizky Januar Akbar, S.Kom., M.Eng.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PELELANGAN ONLINE (E-AUCTION) BERBASIS PERANGKAT BERGERAK ANDROID

ANDRE SETIAWAN
NRP. 5113 100 013

Dosen Pembimbing 1
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing 2
Rizky Januar Akbar, S.Kom., M.Eng.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

(Halaman ini sengaja dikosongkan)



FINAL PROJECT - KI141502

DESIGN AND IMPLEMENTATION OF E-AUCTION MOBILE APPLICATION BASED ON ANDROID.

ANDRE SETIAWAN
NRP. 5113 100 013

Supervisor 1
Rully Soelaiman, S.Kom., M.Kom.

Supervisor 2
Rizky Januar Akbar, S.Kom., M.Eng.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN
RANCANG BANGUN APLIKASI PELELANGAN ONLINE
(E-AUCTION) BERBASIS PERANGKAT BERGERAK
ANDROID

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Dasar dan Terapan Komputasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
ANDRE SETIAWAN
NRP. 5113100013

Disetujui oleh Pembimbing Tugas Akhir:

1. **Rully Soelaiman, S.Kom, M.Kom.**
NIP. 197002131994021001 (Pembimbing 1)
2. **Rizky Januar Akbar, S.Kom, M.Eng.**
NIP. 198701032014041001 (Pembimbing 2)


DEPARTEMEN
TEKNIK INFORMATIKA
SURABAYA
JUNI 2017

(Halaman ini sengaja dikosongkan)

RANCANG BANGUN APLIKASI PELELANGAN ONLINE (E-AUCTION) BERBASIS PERANGKAT BERGERAK ANDROID

Nama : Andre Setiawan
NRP : 5113100013
Jurusan : Teknik Informatika
Fakultas Teknologi Informasi ITS
Dosen Pembimbing I : Rully Soelaiman, S.Kom., M.Kom.
Dosen Pembimbing II : Rizky Januar Akbar, S.Kom., M.Eng.

ABSTRAK

Kemajuan internet di Indonesia mengakibatkan berkembangnya industri e-commerce. Menjamurnya industri e-commerce semakin mempermudah masyarakat Indonesia untuk menjual, mencari dan membeli barang yang diinginkan. Salah satu jenis transaksi jual beli yang telah diketahui secara umum adalah transaksi lelang. Namun, banyak pengguna yang merasa tidak puas dengan sistem lelang online karena banyak hal, seperti yang dijabarkan pada paper “Online auction service failures in Taiwan: Typologies and recovery strategies”[1] yang membahas mengenai kegagalan aplikasi lelang online berdasarkan survey pengguna di Taiwan. Berangkat dari masalah yang diangkat dari paper rujukan dan besarnya pengguna ponsel pintar Android di Indonesia, penulis mengembangkan aplikasi lelang online berbasis perangkat Android dengan fitur tambahan yang disarankan pada paper rujukan. Penulis juga menganalisa UI-UX, fitur, dan alur kerja dari aplikasi Android milik e-commerce besar di Indonesia sehingga aplikasi yang dikembangkan dapat sesuai dengan alur jual beli online yang terjadi di Indonesia. Dengan aplikasi ini, diharapkan dapat meningkatkan peluang lelang online untuk bisa bersaing pada pasar e-commerce di Indonesia.

Kata kunci: *lelang, e-commerce, marketplace.*

(Halaman ini sengaja dikosongkan)

DESIGN AND IMPLEMENTATION OF E-AUCTION MOBILE APPLICATION BASED ON ANDROID

Name : Andre Setiawan
NRP : 5113100013
Department : Department of Informatics
Faculty of Information Technology ITS
Supervisor I : Rully Soelaiman, S.Kom., M.Kom
Supervisor II : Rizky Januar Akbar, S.Kom., M.Eng.

ABSTRACT

Internet progress in Indonesia has resulted in the development of e-commerce industry. The proliferation of e-commerce industry makes it easier for the people of Indonesia to sell. Search and buy the desired goods. One type of sale and purchase transactions that have been known in general is auction transaction. However, many users are dissatisfied with the online auction system for many reasons, as described in the "Online auction service failures in Taiwan: Typologies and recovery strategies" [1] paper that discusses the failure of online auction applications based on user surveys in Taiwan. Departing from the issues raised from reference papers and the large number of Android smartphone users in Indonesia, the author developed an online auction application based on Android devices with additional features suggested on reference papers. The author also analyzes UI-UX, features, and workflow of Android app owned by big e-commerce in Indonesia so that the developed application can be in accordance with online buying and selling process that happened in Indonesia. With this application, the author expects to make online auction opportunities to be able to compete in the e-commerce market in Indonesia.

Keyword: *auction, e-commerce, marketplace.*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan YME karena atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

“Rancang Bangun Aplikasi Pelelangan Online (E-Auction) Berbasis Perangkat Bergerak Android”

Tugas akhir ini dilakukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama proses pengerjaan tugas akhir ini hingga selesai, antara lain:

1. Tuhan YME atas segala karunia dan rahmat-Nya yang telah diberikan selama ini.
2. Orang tua, saudara serta keluarga penulis yang tiada henti-hentinya memberikan semangat, perhatian dan doa selama perkuliahan penulis di Jurusan Teknik Informatika ini.
3. Bapak Rully Soelaiman, S.Kom., M.Kom. dan Bapak Rizky Januar Akbar, S.Kom., M.Eng. selaku dosen pembimbing yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.
4. Teman-teman yaitu Peni Sriwahyu N, Maulina Nur Istiqomah, Ronauli, dan pihak-pihak yang telah banyak membantu dengan memberikan semangat, masukan, bantuan, dan dukungan selama masa pengerjaan tugas akhir.
5. Seluruh pihak yang tidak bisa saya sebutkan satu persatu yang telah memberikan dukungan selama saya menyelesaikan tugas akhir ini.

Mohon maaf apabila terdapat kekurangan dalam penulisan buku tugas akhir ini. Kritik dan saran sangat diharapkan untuk perbaikan dan pembelajaran di kemudian hari. Semoga tugas akhir ini dapat memberikan manfaat yang sebaik-baiknya.

Surabaya, Juni 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xxi
DAFTAR KODE SUMBER.....	xxxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1 Analisa pada Studi Literatur.....	7
2.1.1 Analisa pada <i>Paper</i> Rujukan.....	7
2.1.2 Proses Dasar Pengembangan Perangkat Lunak...	10
2.1.3 Panduan Menyusun User Experience Aplikasi ...	10
2.1.4 Potensi Pasar dari Sistem Operasi Android.....	11
2.2 Analisa Penulis.....	12
2.2.1 Analisa terhadap Aplikasi <i>E-commerce</i> Umum ..	12
2.2.2 Analisa terhadap Aplikasi Serupa	14
2.2.3 Analisa Potensi Aplikasi.....	14
2.2.4 Analisa Keamanan Sistem.....	15
2.2.5 Analisa Arsitektur Sistem.....	15

2.3	Lelang	16
2.4	<i>Online Marketplace</i>	17
2.5	Java	17
2.6	JavaScript	17
2.7	JavaScript Object Notation	18
2.8	Android	18
2.9	Android Studio	19
2.10	<i>Structured Query Language (SQL)</i>	19
2.11	NoSQL	19
2.12	PostgreSQL	20
2.13	Amazon AWS S3	20
2.14	ElasticSearch	20
2.15	MongoDB	21
BAB III ANALISIS DAN PERANCANGAN SISTEM		23
3.1	Analisis Sistem	23
3.1.1	Analisis Kebutuhan	23
3.1.2	Deskripsi Umum Perangkat Lunak	24
3.1.3	Spesifikasi Kebutuhan Aplikasi	25
3.1.4	Identifikasi Pengguna	27
3.2	Perancangan Sistem	27
3.2.1	Aliran Informasi	27
3.2.2	Deskripsi Proses	29
3.2.3	Deskripsi Data	35
3.2.4	Pemodelan Data	37
3.2.5	Perancangan Basis Data	40
3.2.6	Perancangan Arsitektur Sistem	43
3.2.7	Perancangan Antarmuka Pemrograman Aplikasi	44
3.2.8	Perancangan Antarmuka Perangkat Lunak	59
BAB IV IMPLEMENTASI		131

4.1	Lingkungan Pembangunan	131
4.1.1	Lingkungan Pembangunan Perangkat Keras	131
4.1.2	Lingkungan Pembangunan Perangkat Lunak	132
4.2	Implementasi API	132
4.2.1	Implementasi API Privat	132
4.2.2	Implementasi API Publik	165
4.3	Implementasi Fitur Aplikasi	177
4.3.1	Implementasi Fitur Halaman Utama	177
4.3.2	Implementasi Fitur <i>Login</i>	180
4.3.3	Implementasi Fitur <i>Register</i>	183
4.3.4	Implementasi Fitur Halaman Profil	186
4.3.5	Implementasi Fitur <i>Edit</i> Profil	188
4.3.6	Implementasi Fitur <i>Edit</i> Alamat	191
4.3.7	Implementasi Fitur Ubah <i>Password</i>	193
4.3.8	Implementasi Fitur Beri <i>Feedback</i>	196
4.3.9	Implementasi Fitur Detail Beri <i>Feedback</i>	198
4.3.10	Implementasi Fitur <i>Feedback</i> Anda	202
4.3.11	Implementasi Fitur Riwayat	204
4.3.12	Implementasi Fitur Detail Riwayat	207
4.3.13	Implementasi Fitur Favorit	210
4.3.14	Implementasi Fitur Gerai	213
4.3.15	Implementasi Fitur Lelang Barang	215
4.3.16	Implementasi Fitur <i>Edit</i> Barang	219
4.3.17	Implementasi Fitur Detail Barang	223
4.3.18	Implementasi Fitur Pencarian	228
4.3.19	Implementasi Fitur Filter Pencarian	230
4.3.20	Implementasi Fitur Kategori	233
4.3.21	Implementasi Fitur Daftar <i>Chat</i>	236
4.3.22	Implementasi Fitur Ruang <i>Chat</i>	239

4.3.23	Implementasi Fitur Daftar Tawaran.....	241
4.3.24	Implementasi Fitur Daftar Blok.....	249
BAB V PENGUJIAN DAN EVALUASI		253
5.1	Lingkungan Pengujian.....	253
5.2	Skenario Pengujian	253
5.2.1	Pengujian Fungsionalitas	254
5.2.2	Pengujian Berbasis Responden.....	276
5.3	Evaluasi Pengujian	277
5.3.1	Evaluasi Pengujian Fungsionalitas	277
5.3.2	Evaluasi Pengujian Berbasis Responden	277
BAB VI KESIMPULAN DAN SARAN.....		301
6.1	Kesimpulan.....	301
6.2	Saran	302
LAMPIRAN A KUISIONER PENGGUNAAN APLIKASI		303
DAFTAR PUSTAKA		309
BIODATA PENULIS		313

DAFTAR GAMBAR

Gambar 2.1 Hasil Survey Insiden Kegagalan dalam Aplikasi Lelang Online yang Dimuat di <i>Paper</i> Rujukan	8
Gambar 2.2 Hasil Survey terhadap Fasilitas Perbaikan pada Aplikasi Lelang Online	9
Gambar 2.3 Riset yang Dilakukan oleh Steven Hooper Mengenai Bagaimana Pengguna Memegang Ponsel.....	11
Gambar 2.4 Pasar Ponsel Pintar di Dunia pada Q4 2015-2016...	12
Gambar 3.1 Diagram Kasus Penggunaan	26
Gambar 3.2 DFD Level 0	28
Gambar 3.3 DFD Level 1	28
Gambar 3.4 Dekomposisi Mengelola Data Profil Pengguna.....	30
Gambar 3.5 Dekomposisi Mengelola Data Barang Lelang	31
Gambar 3.6 Dekomposisi Mengelola Aktivitas saat Lelang Berlangsung.....	32
Gambar 3.7 Dekomposisi Menjalankan Rangkaian Aktivitas Pengguna	34
Gambar 3.8 <i>Conceptual Data Model</i>	38
Gambar 3.9 <i>Physical Data Model</i>	39
Gambar 3.10 Perancangan Arsitektur Sistem Lelangapa	43
Gambar 3.11 Rancangan Antarmuka Halaman Utama	59
Gambar 3.12 Rancangan Antarmuka Halaman <i>Login</i>	60
Gambar 3.13 Rancangan Antarmuka Halaman Register.....	61
Gambar 3.14 Rancangan Antarmuka Halaman Profil.....	62
Gambar 3.15 Rancangan Antarmuka Halaman <i>Edit</i> Profil	63
Gambar 3.16 Rancangan Antarmuka Halaman <i>Edit</i> Alamat.....	64
Gambar 3.17 Rancangan Antarmuka Halaman Ubah Password .	65
Gambar 3.18 Rancangan Antarmuka Halaman Beri <i>Feedback</i> ...	66
Gambar 3.19 Rancangan Antarmuka Halaman Detail Beri <i>Feedback</i>	67

Gambar 3.20 Rancangan Antarmuka Halaman <i>Feedback</i> Anda.	68
Gambar 3.21 Rancangan Antarmuka Halaman Riwayat	69
Gambar 3.22 Rancangan Antarmuka Halaman Detail Riwayat ..	70
Gambar 3.23 Rancangan Antarmuka Halaman Favorit.....	71
Gambar 3.24 Rancangan Antarmuka Halaman Gerai	72
Gambar 3.25 Rancangan Antarmuka Halaman Lelang Barang...	73
Gambar 3.26 Rancangan Antarmuka Halaman <i>Edit</i> Barang	75
Gambar 3.27 Rancangan Antarmuka Halaman Detail Barang untuk Pengguna Umum	76
Gambar 3.28 Rancangan Antarmuka Halaman Detail Barang untuk Pelelang	77
Gambar 3.29 Rancangan Antarmuka Halaman Pencarian	78
Gambar 3.30 Rancangan Antarmuka Halaman Penyaringan Pencarian	79
Gambar 3.31 Rancangan Antarmuka Halaman Kategori	80
Gambar 3.32 Rancangan Antarmuka Halaman Daftar Chat	81
Gambar 3.33 Rancangan Antarmuka Halaman Ruang Chat	82
Gambar 3.34 Rancangan Antarmuka Halaman Daftar Tawaran.	83
Gambar 3.35 Rancangan Antarmuka Halaman Daftar Blok	84
Gambar 4.1 Halaman Utama	180
Gambar 4.2 Halaman <i>Login</i>	182
Gambar 4.3 Halaman <i>Register</i>	186
Gambar 4.4 Halaman Profil	188
Gambar 4.5 Halaman <i>Edit</i> Profil	191
Gambar 4.6 Halaman <i>Edit</i> Alamat	193
Gambar 4.7 Halaman Ubah <i>Password</i>	196
Gambar 4.8 Halaman Beri <i>Feedback</i> untuk Pelelang.....	198
Gambar 4.9 Halaman Detail Beri <i>Feedback</i>	201
Gambar 4.10 Halaman <i>Feedback</i> Anda.....	203
Gambar 4.11 Halaman Riwayat.....	206
Gambar 4.12 Halaman Detail Riwayat.....	210

Gambar 4.13 Halaman Favorit	212
Gambar 4.14 Halaman Gerai	214
Gambar 4.15 Halaman Lelang Barang	218
Gambar 4.16 Halaman <i>Edit</i> Barang	223
Gambar 4.17 Halaman Detail Barang versi I	227
Gambar 4.18 Halaman Detail Barang versi II	227
Gambar 4.19 Halaman Pencarian	230
Gambar 4.20 Halaman Penyaringan Pencarian	233
Gambar 4.21 Halaman Kategori	236
Gambar 4.22 Halaman Daftar Chat	239
Gambar 4.23 Halaman Ruang Chat	241
Gambar 4.24 Halaman Daftar Tawaran saat Lelang Berlangsung	245
Gambar 4.25 Halaman Daftar Tawaran saat Lelang Selesai	248
Gambar 4.26 Halaman Daftar Blok	251
Gambar 5.1 Skenario 1. Mengubah Data Profil	255
Gambar 5.2 Skenario 2. Mengubah Data Alamat	255
Gambar 5.3 Skenario 3. Mengubah Data Password	256
Gambar 5.4 Skenario 1. Menambah Barang Baru	257
Gambar 5.5 Skenario 1. Memperbaharui Data Barang	258
Gambar 5.6 Skenario 1.1. Memasukkan Penawaran Baru (Kondisi Ditolak)	259
Gambar 5.7 Skenario 1.2. Memasukkan Penawaran Baru (Kondisi Diterima)	259
Gambar 5.8 Skenario 2.1. Memilih Tawaran Sebagai Pemenang (Melalui Halaman Detail)	260
Gambar 5.9 Skenario 2.2. Memilih Tawaran Sebagai Pemenang (Melalui Halaman Daftar Tawaran)	261
Gambar 5.10 Skenario 3. Membatalkan Tawaran dari Penawar	262
Gambar 5.11 Skenario 4. Melakukan Blok pada Penawar	263
Gambar 5.12 Skenario 5. Menampilkan Daftar Blok	263

Gambar 5.13 Skenario 6. Membatalkan Lelang	264
Gambar 5.14 Skenario 1. Menampilkan Ulasan (Alternatif 1) ..	265
Gambar 5.15 Skenario 1. Menampilkan Ulasan (Alternatif 2) ..	265
Gambar 5.16 Skenario 2. Memberikan Ulasan kepada Pemenang	266
Gambar 5.17 Skenario 3. Memberikan Ulasan kepada Pelelang	267
Gambar 5.18 Skenario 4. Mengubah Ulasan yang Telah Diberikan kepada Pemenang	268
Gambar 5.19 Skenario 5. Mengubah Ulasan yang Telah Diberikan kepada Pelelang	269
Gambar 5.20 Skenario 1. Melakukan Pencarian	270
Gambar 5.21 Skenario 1. Menampilkan Daftar Riwayat	271
Gambar 5.22 Skenario 2. Menampilkan Detail Riwayat	271
Gambar 5.23 Skenario 1. Menampilkan Notifikasi Masuk	272
Gambar 5.24 Skenario 1. Fitur Pesan Singkat (Alternatif 1)	273
Gambar 5.25 Skenario 1. Fitur Pesan Singkat (Alternatif 2)	274
Gambar 5.26 Skenario 1. Memasukkan Barang Favorit	275
Gambar 5.27 Skenario 2. Menghapus Barang Favorit (Alternatif 1)	275
Gambar 5.28 Skenario 2. Menghapus Barang Favorit (Alternatif 2)	276

DAFTAR TABEL

Tabel 3.1 Kualitas Perangkat Lunak	27
Tabel 3.2 Deskripsi Data (1)	36
Tabel 3.3 Deskripsi Data (2)	37
Tabel 3.4 Daftar Atribut Tabel <i>Users</i>	85
Tabel 3.5 Daftar Atribut Tabel <i>Items</i> (1).....	86
Tabel 3.6 Daftar Atribut Tabel <i>Items</i> (2).....	87
Tabel 3.7 Daftar Atribut Tabel <i>Bids</i>	87
Tabel 3.8 Daftar Atribut Tabel <i>Bidlogs</i>	88
Tabel 3.9 Daftar Atribut Tabel <i>BiddingBlock</i>	88
Tabel 3.10 Daftar Atribut Tabel <i>Categories</i>	89
Tabel 3.11 Daftar Atribut Tabel <i>Cities</i>	89
Tabel 3.12 Daftar Atribut Tabel <i>Provinces</i>	89
Tabel 3.13 Daftar Atribut Tabel <i>RatingLogs</i>	90
Tabel 3.14 Daftar Atribut Tabel <i>RatingAuctioneers</i>	91
Tabel 3.15 Daftar Atribut Tabel <i>RatingBidders</i>	92
Tabel 3.16 Daftar Atribut Tabel <i>Notification</i>	93
Tabel 3.17 Daftar Atribut Tabel <i>Favorites</i>	93
Tabel 3.18 Daftar Atribut Dokumen <i>ItemImages</i>	94
Tabel 3.19 Daftar Atribut Dokumen <i>ChatRoom</i>	94
Tabel 3.20 Daftar Atribut Dokumen <i>UserChat</i>	95
Tabel 3.21 <i>Request</i> API Modul Informasi Pengguna	95
Tabel 3.22 <i>Response Body</i> API Modul Informasi Pengguna Bagian 1	95
Tabel 3.23 <i>Response Body</i> API Modul Informasi Pengguna Bagian 2	96
Tabel 3.24 <i>Request</i> Body API Modul Ubah Profil Pengguna	96
Tabel 3.25 <i>Response Body</i> API Modul Ubah Profil Pengguna ...	96
Tabel 3.26 <i>Request</i> API Modul Ubah Alamat Pengguna	97
Tabel 3.27 <i>Response Body</i> API Modul Ubah Alamat Pengguna	97

Tabel 3.28 <i>Request</i> API Modul Ubah Password Pengguna.....	97
Tabel 3.29 <i>Response Body</i> API Modul Ubah Password Pengguna	97
Tabel 3.30 <i>Request</i> API Modul Informasi Barang	98
Tabel 3.31 <i>Response Body</i> API Modul Informasi Barang.....	98
Tabel 3.32 <i>Response Body</i> API Modul Informasi Barang.....	98
Tabel 3.33 <i>Request</i> API Modul Lelang Barang.....	99
Tabel 3.34 <i>Response Body</i> API Modul Lelang Barang	99
Tabel 3.35 <i>Request</i> API Modul Lelang Barang.....	100
Tabel 3.36 <i>Response Body</i> API Modul Lelang Barang	100
Tabel 3.37 <i>Request</i> API Modul Hapus Barang.....	100
Tabel 3.38 <i>Response Body</i> API Modul Hapus Barang	101
Tabel 3.39 <i>Request</i> API Modul Mendapatkan Daftar Barang Milik Pengguna	101
Tabel 3.40 <i>Response Body</i> API Modul Mendapatkan Daftar Barang Milik Pengguna (Bagian 1)	101
Tabel 3.41 <i>Response Body</i> API Modul Mendapatkan Daftar Barang Milik Pengguna (Bagian 1)	101
Tabel 3.42 <i>Request</i> API Modul Mendapatkan Daftar Ulasan untuk Pelelang	102
Tabel 3.43 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan untuk Pelelang (Bagian 1)	102
Tabel 3.44 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan untuk Pelelang (Bagian 2)	102
Tabel 3.45 <i>Request</i> API Modul Mendapatkan Daftar Ulasan sebagai Pemenang	102
Tabel 3.46 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan sebagai Pemenang	103
Tabel 3.47 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan sebagai Pemenang	103

Tabel 3.48 <i>Request</i> API Modul Mendapatkan Daftar Ulasan yang Dapat Diberikan kepada Pemenang.....	103
Tabel 3.49 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan yang Dapat Diberikan kepada Pemenang.....	103
Tabel 3.50 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan yang Dapat Diberikan kepada Pemenang.....	104
Tabel 3.51 <i>Request</i> API Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang	104
Tabel 3.52 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang	104
Tabel 3.53 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang	105
Tabel 3.54 <i>Request</i> API Modul Memasukkan Ulasan Baru untuk Pemenang	105
Tabel 3.55 <i>Response Body</i> API Modul Informasi Pengg Memasukkan Ulasan Baru untuk Pemenang una	105
Tabel 3.56 <i>Request</i> API Modul Memasukkan Ulasan Baru untuk Pelelang	106
Tabel 3.57 <i>Response Body</i> API Modul Memasukkan Ulasan Baru untuk Pelelang	106
Tabel 3.58 <i>Request</i> API Modul Memperbaharui Ulasan untuk Pemenang	106
Tabel 3.59 <i>Response Body</i> API Modul Memperbaharui Ulasan untuk Pemenang	107
Tabel 3.60 <i>Request</i> API Modul Memperbaharui Ulasan untuk Pelelang	107
Tabel 3.61 <i>Response Body</i> API Modul Memperbaharui Ulasan untuk Pelelang	107
Tabel 3.62 <i>Request</i> API Modul Mendapatkan Detail Ulasan untuk Pemenang	107

Tabel 3.63 <i>Response Body</i> API Modul Mendapatkan Detail Ulasan untuk Pemenang	108
Tabel 3.64 <i>Response Body</i> API Modul Mendapatkan Detail Ulasan untuk Pemenang	108
Tabel 3.65 <i>Request</i> API Modul Mendapatkan Detail Ulasan untuk Pelelang	108
Tabel 3.66 <i>Response Body</i> API Modul Mendapatkan Detail Ulasan untuk Pelelang	109
Tabel 3.67 <i>Response Body</i> API Modul Mendapatkan Detail Ulasan untuk Pelelang	109
Tabel 3.68 <i>Request</i> API Modul Mendapatkan Seluruh Daftar Riwayat Penawaran	109
Tabel 3.69 <i>Response Body</i> API Modul Mendapatkan Seluruh Daftar Riwayat Penawaran	110
Tabel 3.70 <i>Response Body</i> API Modul Mendapatkan Seluruh Daftar Riwayat Penawaran	110
Tabel 3.71 <i>Request</i> API Modul Mendapatkan Detail suatu Riwayat Penawaran.....	110
Tabel 3.72 <i>Response Body</i> API Modul Mendapatkan Detail suatu Riwayat Penawaran	111
Tabel 3.73 <i>Response Body</i> API Modul Mendapatkan Detail suatu Riwayat Penawaran	111
Tabel 3.74 <i>Request</i> API Modul Mendapatkan Status Favorit suatu Barang	111
Tabel 3.75 <i>Response Body</i> API Modul Mendapatkan Status Favorit suatu Barang	112
Tabel 3.76 <i>Response Body</i> API Modul Mendapatkan Status Favorit suatu Barang	112
Tabel 3.77 <i>Request</i> API Modul Mendapatkan Daftar Barang Favorit	112

Tabel 3.78 <i>Response Body</i> API Modul Mendapatkan Daftar Barang Favorit	112
Tabel 3.79 <i>Response Body</i> API Modul Mendapatkan Daftar Barang Favorit	113
Tabel 3.80 <i>Request</i> API Modul Memasukkan Barang sebagai Favorit	113
Tabel 3.81 <i>Response Body</i> API Modul Memasukkan Barang sebagai Favorit	113
Tabel 3.82 <i>Request</i> API Modul Menghapus Barang dari Favorit	114
Tabel 3.83 <i>Response Body</i> API Modul Menghapus Barang dari Favorit	114
Tabel 3.84 <i>Request</i> API Modul Mendapatkan Daftar Ruang <i>Chat</i>	114
Tabel 3.85 <i>Response Body</i> API Modul Mendapatkan Daftar Ruang <i>Chat</i>	114
Tabel 3.86 <i>Response Body</i> API Modul Mendapatkan Daftar Ruang <i>Chat</i>	115
Tabel 3.87 <i>Request</i> API Modul Mendapatkan Daftar Tawaran suatu Barang.....	115
Tabel 3.88 <i>Response Body</i> API Modul Mendapatkan Daftar Tawaran suatu Barang.....	115
Tabel 3.89 <i>Response Body</i> API Modul Mendapatkan Daftar Tawaran suatu Barang.....	116
Tabel 3.90 <i>Request</i> API Modul Memasukkan Token FCM.....	116
Tabel 3.91 <i>Response Body</i> API Modul Memasukkan Token FCM	116
Tabel 3.92 <i>Request</i> API Modul Mengubah Status Login ketika <i>Logout</i>	116
Tabel 3.93 <i>Response Body</i> API Modul Mengubah Status Login ketika <i>Logout</i>	116

Tabel 3.94 <i>Request</i> API Modul Memblok Pengguna	117
Tabel 3.95 <i>Response Body</i> API Modul Memblok Pengguna	117
Tabel 3.96 <i>Request</i> API Modul Membebaskan Pengguna	117
Tabel 3.97 <i>Response Body</i> API Modul Membebaskan Pengguna	117
Tabel 3.98 <i>Request</i> API Modul Mendapatkan Daftar Blok.....	118
Tabel 3.99 <i>Response Body</i> API Modul Mendapatkan Daftar Blok	118
Tabel 3.100 <i>Response Body</i> API Modul Mendapatkan Daftar Blok	118
Tabel 3.101 <i>Request</i> API Modul Memilih Pemenang Lelang ...	118
Tabel 3.102 <i>Response Body</i> API Modul Memilih Pemenang Lelang	118
Tabel 3.103 <i>Request</i> API Modul Melakukan Validasi Token ...	119
Tabel 3.104 <i>Response Body</i> API Modul Melakukan Validasi Token	119
Tabel 3.105 <i>Request</i> API Modul Mendapatkan Daftar Barang.	119
Tabel 3.106 <i>Response Body</i> API Modul Mendapatkan Daftar Barang	119
Tabel 3.107 <i>Response Body</i> API Modul Mendapatkan Daftar Barang	120
Tabel 3.108 <i>Request</i> API Modul Login.....	120
Tabel 3.109 <i>Response Body</i> API Modul Login	120
Tabel 3.110 <i>Response Body</i> API Modul Login	121
Tabel 3.111 <i>Request</i> API Modul Register	121
Tabel 3.112 <i>Response Body</i> API Modul Register	121
Tabel 3.113 <i>Request</i> API Modul Melakukan Validasi Input <i>Username</i>	121
Tabel 3.114 <i>Response Body</i> API Modul Melakukan Validasi Input <i>Username</i>	122

Tabel 3.115 <i>Request</i> API Modul Melakukan Validasi Input <i>Domain</i>	122
Tabel 3.116 <i>Response Body</i> API Modul Melakukan Validasi Input <i>Domain</i>	122
Tabel 3.117 <i>Request</i> API Modul Mendapatkan Daftar Ulasan sebagai Pemenang	122
Tabel 3.118 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan sebagai Pemenang	122
Tabel 3.119 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan sebagai Pemenang	123
Tabel 3.120 <i>Request</i> API Modul Mendapatkan Daftar Ulasan sebagai Pelelang	123
Tabel 3.121 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan sebagai Pelelang	123
Tabel 3.122 <i>Response Body</i> API Modul Mendapatkan Daftar Ulasan sebagai Pelelang	124
Tabel 3.123 <i>Request</i> API Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang	124
Tabel 3.124 <i>Response Body</i> API Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang	124
Tabel 3.125 <i>Request</i> API Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang	124
Tabel 3.126 <i>Response Body</i> API Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang	125
Tabel 3.127 <i>Request</i> API Modul Mendapatkan Daftar Riwayat Penawaran	125
Tabel 3.128 <i>Response Body</i> API Modul Mendapatkan Daftar Riwayat Penawaran	125
Tabel 3.129 <i>Response Body</i> API Modul Mendapatkan Daftar Riwayat Penawaran	126
Tabel 3.130 <i>Request</i> API Modul Mendapatkan Detail Barang	126

Tabel 3.131 <i>Response Body</i> API Modul Mendapatkan Detail Barang	126
Tabel 3.132 <i>Response Body</i> API Modul Mendapatkan Detail Barang	127
Tabel 3.133 <i>Request</i> API Modul Mendapatkan Peringkat Penawaran pada Detail Barang.....	127
Tabel 3.134 <i>Response Body</i> API Modul Mendapatkan Peringkat Penawaran pada Detail Barang.....	128
Tabel 3.135 <i>Response Body</i> API Modul Mendapatkan Peringkat Penawaran pada Detail Barang.....	128
Tabel 3.136 <i>Request</i> API Modul Pencarian berdasarkan Kata Kunci	128
Tabel 3.137 <i>Request</i> API Modul Pencarian berdasarkan Penyaringan	129
Tabel 3.138 <i>Request</i> API Modul Menampilkan semua Barang pada suatu Kategori.....	129
Tabel 3.139 <i>Response Body</i> API Modul Pencarian	129
Tabel 3.140 <i>Response Body</i> API Modul Pencarian	129
Tabel 3.141 <i>Response Body</i> API Modul Pencarian	130
Tabel 5.1 Pengujian Fungsionalitas Fitur Mengelola Data Profil Pengguna (1).....	278
Tabel 5.2 Pengujian Fungsionalitas Fitur Mengelola Data Profil Pengguna (2).....	279
Tabel 5.3 Pengujian Fungsionalitas Fitur Mengelola Data Barang Lelang (1)	280
Tabel 5.4 Pengujian Fungsionalitas Fitur Mengelola Data Barang Lelang (2)	281
Tabel 5.5 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (1)	282
Tabel 5.6 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (2)	283

Tabel 5.7 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (3).....	284
Tabel 5.8 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (4).....	285
Tabel 5.9 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (5).....	286
Tabel 5.10 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (6).....	287
Tabel 5.11 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (1).....	288
Tabel 5.12 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (2).....	289
Tabel 5.13 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (3).....	290
Tabel 5.14 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (4).....	291
Tabel 5.15 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (5).....	292
Tabel 5.16 Pengujian Fungsionalitas Fitur Menampilkan Proses Pencarian Barang.....	292
Tabel 5.17 Pengujian Fungsionalitas Fitur Menampilkan Daftar dan Detail Riwayat Penawaran (1).....	293
Tabel 5.18 Pengujian Fungsionalitas Fitur Menampilkan Daftar dan Detail Riwayat Penawaran (2).....	294
Tabel 5.19 Pengujian Fungsionalitas Fitur Menampilkan Notifikasi Masuk.....	294
Tabel 5.20 Pengujian Fungsionalitas Fitur Berkirim Pesan Singkat	295
Tabel 5.21 Pengujian Fungsionalitas Fitur Mengelola Barang Favorit (1).....	296

Tabel 5.22 Pengujian Fungsionalitas Fitur Mengelola Barang Favorit (2).....297

Tabel 5.23 Rangkuman Hasil Pengujian Fungsionalitas (1)298

Tabel 5.24 Rangkuman Hasil Pengujian Fungsionalitas (2)299

Tabel 5.25 Rangkuman Hasil Kuisisioner Kepuasan Pengguna ..300

Tabel A.1. Kuisisioner Penggunaan Aplikasi M. Luthfie La Roeha303

Tabel A.2. Kuisisioner Penggunaan Aplikasi Billy304

Tabel A.3. Kuisisioner Penggunaan Aplikasi Johanes Andre.....305

Tabel A.4. Kuisisioner Penggunaan Aplikasi Freddy Hermawan306

Tabel A.5. Kuisisioner Penggunaan Aplikasi Albert Bungaran...307

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Modul Informasi Pengguna ...	133
Kode Sumber 4.2 Implementasi Modul Ubah Profil Pengguna	134
Kode Sumber 4.3 Implementasi Modul Ubah Alamat Pengguna	134
Kode Sumber 4.4 Implementasi Modul Ubah Password Pengguna	135
Kode Sumber 4.5 Implementasi Modul Informasi Barang	137
Kode Sumber 4.6 Implementasi Modul Lelang Barang	138
Kode Sumber 4.7 Implementasi Modul <i>Edit</i> Lelang Barang	139
Kode Sumber 4.8 Implementasi Modul Hapus Barang	140
Kode Sumber 4.9 Implementasi Modul Hapus Barang	141
Kode Sumber 4.10 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pelelang	142
Kode Sumber 4.11 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pemenang	143
Kode Sumber 4.12 Implementasi Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pemenang	144
Kode Sumber 4.13 Implementasi Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang	145
Kode Sumber 4.14 Implementasi Modul Memasukkan Ulasan Baru untuk Pemenang	146
Kode Sumber 4.15 Implementasi Modul Memasukkan Ulasan Baru untuk Pelelang	147
Kode Sumber 4.16 Implementasi Modul Memperbaharui Ulasan untuk Pemenang	148
Kode Sumber 4.17 Implementasi Modul Memperbaharui Ulasan untuk Pelelang	149
Kode Sumber 4.18 Implementasi Modul Mendapatkan Detail Ulasan untuk Pemenang	150

Kode Sumber 4.19 Implementasi Modul Mendapatkan Detail Ulasan untuk Pelelang	151
Kode Sumber 4.20 Implementasi Modul Mendapatkan Seluruh Daftar Riwayat Penawaran	152
Kode Sumber 4.21 Implementasi Modul Mendapatkan Detail suatu Riwayat Penawaran	152
Kode Sumber 4.22 Implementasi Modul Mendapatkan Status Favorit suatu Barang.....	153
Kode Sumber 4.23 Implementasi Modul Daftar Barang Favorit	154
Kode Sumber 4.24 Implementasi Modul Memasukkan Barang sebagai Favorit.....	155
Kode Sumber 4.25 Implementasi Modul Menghapus Barang dari Favorit	156
Kode Sumber 4.26 Implementasi Modul Mendapatkan Daftar Ruang <i>Chat</i>	158
Kode Sumber 4.27 Implementasi Modul Mendapatkan Daftar Tawaran suatu Barang	159
Kode Sumber 4.28 Implementasi Modul Memasukkan Token FCM	160
Kode Sumber 4.29 Implementasi Modul Mengubah Status Login ketika <i>Logout</i>	161
Kode Sumber 4.30 Implementasi Modul Memblok Pengguna .	162
Kode Sumber 4.31 Implementasi Modul Membebaskan Pengguna	163
Kode Sumber 4.32 Implementasi Modul Mendapatkan Daftar Blok	163
Kode Sumber 4.33 Implementasi Modul Memilih Pemenang Lelang.....	164
Kode Sumber 4.34 Implementasi Modul Melakukan Validasi Token.....	165

Kode Sumber 4.35 Implementasi Modul Mendapatkan Daftar Barang	166
Kode Sumber 4.36 Implementasi Modul Login	167
Kode Sumber 4.37 Implementasi Modul Register	167
Kode Sumber 4.38 Implementasi Modul Melakukan Validasi Input Username.....	168
Kode Sumber 4.39 Implementasi Modul Melakukan Validasi Input Domain	169
Kode Sumber 4.40 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pemenang	169
Kode Sumber 4.41 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pelelang	170
Kode Sumber 4.42 Implementasi Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang	171
Kode Sumber 4.43 Implementasi Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang	172
Kode Sumber 4.44 Implementasi Modul Mendapatkan Daftar Riwayat Penawaran	173
Kode Sumber 4.45 Implementasi Modul Mendapatkan Detail Barang	174
Kode Sumber 4.46 Implementasi Modul Informasi Pengguna	174
Kode Sumber 4.47 Implementasi Modul Pencarian berdasarkan Kata Kunci.....	175
Kode Sumber 4.48 Implementasi Modul Pencarian berdasarkan Penyaringan	176
Kode Sumber 4.49 Implementasi Modul Menampilkan semua Barang pada suatu Kategori	177
Kode Sumber 4.50 Menampilkan Muatan Halaman Utama.....	179
Kode Sumber 4.51 Menampilkan Halaman Utama.....	179
Kode Sumber 4.52 Menampilkan Muatan Halaman <i>Login</i>	181
Kode Sumber 4.53 Menampilkan Halaman <i>Login</i>	182

Kode Sumber 4.54 Menampilkan Muatan Halaman <i>Register</i> ...	183
Kode Sumber 4.55 Menampilkan Halaman <i>Register</i>	185
Kode Sumber 4.56 Menampilkan Muatan Halaman Profil	187
Kode Sumber 4.57 Menampilkan Halaman Profil	187
Kode Sumber 4.58 Menampilkan Muatan Halaman <i>Edit</i> Profil	189
Kode Sumber 4.59 Menampilkan Halaman <i>Edit</i> Profil.....	190
Kode Sumber 4.60 Menampilkan Muatan Halaman <i>Edit</i> Alamat	192
Kode Sumber 4.61 Menampilkan Halaman <i>Edit</i> Alamat	192
Kode Sumber 4.62 Menampilkan Muatan Halaman Ubah <i>Password</i>	194
Kode Sumber 4.63 Menampilkan Halaman Ubah <i>Password</i>	195
Kode Sumber 4.64 Menampilkan Muatan Halaman Beri <i>Feedback</i> untuk Pemenang	197
Kode Sumber 4.65 Menampilkan Halaman Beri <i>Feedback</i> untuk Pemenang	197
Kode Sumber 4.66 Menampilkan Muatan Halaman Detail Beri <i>Feedback</i>	199
Kode Sumber 4.67 Menampilkan Halaman Detail Beri <i>Feedback</i>	201
Kode Sumber 4.68 Menampilkan Muatan Halaman <i>Feedback</i> Anda	203
Kode Sumber 4.69 Menampilkan Halaman <i>Feedback</i> Anda	203
Kode Sumber 4.70 Menampilkan Muatan Halaman Riwayat ...	205
Kode Sumber 4.71 Menampilkan Halaman Riwayat	205
Kode Sumber 4.72 Menampilkan Muatan Halaman Detail Riwayat	208
Kode Sumber 4.73 Menampilkan Halaman Detail Riwayat	209
Kode Sumber 4.74 Menampilkan Muatan Halaman Favorit.....	211
Kode Sumber 4.75 Menampilkan Halaman Favorit	212
Kode Sumber 4.76 Menampilkan Muatan Halaman Gerai	213

Kode Sumber 4.77 Menampilkan Halaman Gerai	214
Kode Sumber 4.78 Menampilkan Muatan Halaman Lelang Barang	215
Kode Sumber 4.79 Menampilkan Halaman Lelang Barang	218
Kode Sumber 4.80 Menampilkan Muatan Halaman <i>Edit</i> Barang	220
Kode Sumber 4.81 Menampilkan Halaman <i>Edit</i> Barang	222
Kode Sumber 4.82 Menampilkan Muatan Halaman Detail Barang versi I.....	225
Kode Sumber 4.83 Menampilkan Halaman Detail Barang versi I dan II	226
Kode Sumber 4.84 Menampilkan Muatan Halaman Detail Barang versi II	226
Kode Sumber 4.85 Menampilkan Muatan Halaman Pencarian	229
Kode Sumber 4.86 Menampilkan Halaman Pencarian.....	229
Kode Sumber 4.87 Menampilkan Muatan Halaman Penyaringan Pencarian	232
Kode Sumber 4.88 Menampilkan Halaman Penyaringan Pencarian	232
Kode Sumber 4.89 Menampilkan Muatan Halaman Kategori ..	234
Kode Sumber 4.90 Menampilkan Halaman Kategori	235
Kode Sumber 4.91 Menampilkan Muatan Halaman Daftar Chat	237
Kode Sumber 4.92 Menampilkan Halaman Daftar Chat.....	238
Kode Sumber 4.93 Menampilkan Muatan Halaman Ruang Chat	240
Kode Sumber 4.94 Menampilkan Halaman Ruang Chat	240
Kode Sumber 4.95 Menampilkan Muatan Halaman Daftar Tawaran saat Lelang Berlangsung	243
Kode Sumber 4.96 Menampilkan Halaman Daftar Tawaran saat Lelang Berlangsung.....	244

Kode Sumber 4.97 Menampilkan Muatan Halaman Daftar Tawaran saat Lelang Selesai.....246

Kode Sumber 4.98 Menampilkan Halaman Daftar Tawaran saat Lelang Selesai.....248

Kode Sumber 4.99 Menampilkan Muatan Halaman Daftar Blok250

Kode Sumber 4.100 Menampilkan Halaman Daftar Blok.....251

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan hal-hal yang menjadi latar belakang, permasalahan yang dihadapi, batasan masalah, tujuan, metodologi dan sistematika penulisan yang digunakan dalam pembuatan buku tugas akhir ini.

1.1 Latar Belakang

Teknologi internet yang sedang berkembang di Indonesia saat ini membuat tumbuhnya berbagai *e-commerce*. *E-commerce* yang sedang tumbuh dan berkembang di Indonesia pun bervariasi pada berbagai *niche* yang disasar oleh pendirinya. Terdapat banyak sekali variasi *e-commerce* saat ini, diantaranya *online marketplace*, pembelian tiket pesawat dan hotel secara *online*, pembelian tiket kereta secara *online*, situs *e-commerce* iklan baris, dan masih banyak lagi.

Namun, masih ada sebuah peluang *e-commerce* yang belum seberapa populer di Indonesia, yaitu sistem *e-auction* atau lelang secara *online*. Seperti yang kita ketahui, sistem lelang biasanya dilakukan oleh instansi-instansi tertentu dan terkesan masih sangat kaku. Lelang adalah penjualan barang secara terbuka untuk umum dengan penawaran harga secara tertulis dan lisan yang semakin meningkat atau menurun untuk mencapai harga tertinggi yang didahului dengan pengumuman lelang.

Lelang konvensional biasanya diadakan oleh sebuah perusahaan yang bergerak di bidang jual beli barang lelang di mana proses lelang diadakan di sebuah tempat sehingga konsumen atau penawar bisa berdatangan. Melalui tugas akhir ini, pelelangan barang dapat dilakukan secara *online* dan siapapun sebagai pengguna terdaftar dapat melelang barang apapun dan menawar barang apapun, layaknya sebuah pasar pelelangan. Selain itu, diharapkan aplikasi lelang *online* yang dibuat dapat menghapus stigma-stigma buruk yang melekat pada sistem lelang *online* yang sudah ada sebelumnya, seperti data yang tidak *reliable*, alur proses

pelelangan yang tidak jelas dan membingungkan, produk yang didapatkan ternyata tidak sesuai dengan informasi pada saat produk dilelang (*bad information*), serta keamanan dari sistem lelang *online* yang kurang diperhatikan oleh pihak pengembang [1].

Hasil dari tugas akhir ini adalah berupa aplikasi sistem *e-auction* atau sistem pasar pelelangan *online* berbasis Android, dimana aplikasi ini dapat menambah *user experience* pengguna dan mempercepat kebutuhan pengguna untuk melakukan pelelangan, cukup melalui aplikasi Android, mengambil foto barang yang dilelang dan mengisi data-data, pengguna dapat langsung melelang barangnya. Dengan adanya aplikasi lelang *online* berbasis Android ini diharapkan dapat mempermudah dan mempercepat kebutuhan pengguna dalam melakukan lelang. Selain itu, melalui tugas akhir ini diharapkan dapat menghasilkan aplikasi *reliable*, jelas, dan aman.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Mendefinisikan kebutuhan aplikasi pelelangan *online* yang akan dibangun (*requirement analysis*) sesuai dengan kebutuhan yang dibutuhkan pengguna;
2. Merancang arsitektur API dan aplikasi pelelangan *online* yang akan dibangun;
3. Melakukan pengembangan API pelelangan *online* dan aplikasi pelelangan *online* yang akan dijalankan pada sistem operasi Android; dan
4. Merencanakan pengujian *black-box* dan pengujian berbasis responden untuk mengukur kepuasan pada aplikasi pelelangan *online* yang telah dibangun.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Aplikasi dibangun di atas sistem operasi Android dan dijalankan pada perangkat ponsel pintar dengan posisi *potrait*;
2. Aplikasi dibangun dengan menggunakan bahasa pemrograman Java (*native*);
3. Aplikasi harus dapat berkomunikasi dengan basis data yang terdapat pada server dengan menggunakan antarmuka pemrograman aplikasi yang telah dibuat;
4. Jangkauan pengerjaan dalam aplikasi ini tidak sampai meliputi proses pembayaran dengan sistem rekening bersama, dikarenakan keterbatasan dana, dan juga tidak meliputi implementasi proses promosi (*promo*);
5. Antarmuka pemrograman aplikasi dibangun dengan menggunakan bahasa pemrograman Javascript; dan
6. Pengujian aplikasi ditujukan kepada kalangan umum.

1.4 Tujuan

Tujuan dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Membuat rancangan kebutuhan aplikasi pelelangan *online* yang akan dibangun;
2. Membangun rancangan arsitektur API dan aplikasi pelelangan *online*;
3. Membangun API dan aplikasi pelelangan *online* yang akan dijalankan pada perangkat bergerak Android; dan
4. Melakukan pengujian *black-box* dan pengujian berbasis responden untuk mengukur kepuasan pada aplikasi pelelangan *online* yang telah dibangun.

1.5 Metodologi

Ada beberapa tahapan dalam pengerjaan tugas akhir ini, yaitu sebagai berikut:

1. Studi Literatur
Pada tahap ini dilakukan pengumpulan informasi mengenai bagaimana prosedur dan aturan lelang secara

umum. Pengumpulan informasi dilakukan untuk menggali kebutuhan dalam menyusun rancangan dan implementasi sistem yang akan dibangun. Selain itu juga dilakukan studi awal mengenai jalannya daur hidup aplikasi Android, dasar-dasar pemrograman aplikasi Android, bahasa pemrograman Java, dan mengenai teknologi yang akan digunakan untuk membangun antarmuka pemrograman aplikasi.

2. Analisis dan perancangan sistem

Pada tahap ini dilakukan analisis dan pendefinisian kebutuhan sistem untuk masalah yang sedang dihadapi. Penelitian dan analisis dilakukan untuk menggali kebutuhan yang ada dalam pelepasan dan *marketplace* yang dapat diimplementasikan kedalam aplikasi. Selanjutnya, dilakukan perancangan sistem dengan tahapan sebagai berikut:

- a. analisis aktor yang terlibat dalam sistem;
- b. perancangan model kasus penggunaan yang merupakan kebutuhan pada elemen yang akan dibangun;
- c. perancangan dan desain arsitektur aplikasi;
- d. perancangan antarmuka aplikasi; dan
- e. perancangan proses di dalam aplikasi.

3. Implementasi

Pada tahap ini dilakukan pembuatan elemen perangkat lunak yang merupakan implementasi dari rancangan yang telah dibuat sebelumnya. Perincian tahap ini adalah sebagai berikut:

- a. implementasi API;
- b. implementasi fitur-fitur utama dalam aplikasi; dan
- c. implementasi fitur-fitur tambahan seperti berkirim pesan singkat (*chatting*), *push*

notification, melihat daftar riwayat, dan melihat dan memasukkan *rating* dan *feedback*.

4. Pengujian dan evaluasi
Tahap pengujian dan evaluasi akan dilakukan dengan menguji aplikasi pada pengguna umum dari berbagai kalangan. Selain itu juga dilakukan pengujian mutu perangkat lunak dan *benchmarking* pada server.
5. Penyusunan buku tugas akhir
Tahap ini merupakan tahap penyusunan laporan berupa buku sebagai dokumentasi pengerjaan tugas akhir yang mencakup seluruh dasar teori, desain, implementasi serta hasil pengujian yang telah dilakukan.

1.6 Sistematika Penulisan

Penulisan buku tugas akhir ini dibagi kedalam 6 bab yang masing-masing menjelaskan bagian-bagian yang berbeda namun tetap memiliki korelasi satu dengan yang lain, yaitu:

1. Bab I, Pendahuluan, berisi penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi serta sistematika penulisan buku.
2. Bab II, Dasar Teori, berisi penjelasan teori-teori yang digunakan sebagai dasar pengerjaan tugas akhir ini.
3. Bab III, Analisis dan Perancangan Sistem, berisi rancangan pembuatan sistem penyelesaian permasalahan dalam tugas akhir ini.
4. Bab IV, Implementasi, berisi lingkungan serta hasil penerapan rancangan sistem penyelesaian permasalahan dalam tugas akhir ini dalam bentuk sumber kode beserta penjelasannya.
5. Bab V, Pengujian dan Evaluasi, berisi lingkungan serta hasil dari rangkaian uji coba yang dilakukan untuk menguji kebenaran serta kinerja dari sistem.
6. Bab VI, Kesimpulan dan Saran, berisi kesimpulan pengerjaan tugas akhir ini dan saran untuk pengembangan kedepannya.

7. Daftar Pustaka, merupakan daftar referensi yang digunakan dalam pengembangan Tugas Akhir.
8. Lampiran, merupakan bab tambahan yang berisi daftar istilah yang penting dalam aplikasi ini.

BAB II

DASAR TEORI

Pada bab ini akan membahas mengenai dasar teori dan literatur yang menjadi dasar pengerjaan tugas akhir ini.

2.1 Analisa pada Studi Literatur

Pada subbab ini akan dibahas mengenai analisa yang terdapat pada studi literatur yang dipilih oleh penulis sebagai referensi dalam pengerjaan tugas akhir ini.

2.1.1 Analisa pada *Paper* Rujukan

Dalam pengerjaan tugas akhir, penulis merujuk pada sebuah paper yang membahas mengenai peran kepuasan pengguna terhadap keberhasilan layanan *e-commerce* lelang *online* dengan melakukan survei terhadap pengguna [1]. Berdasarkan paper tersebut didapatkan bahwa terdapat berbagai faktor yang menyebabkan pengguna tidak cukup puas dengan pelayanan yang diberikan oleh layanan yang sudah ada. Selain itu, paper juga memberikan solusi-solusi yang diberikan oleh pengguna aplikasi untuk memperbaiki kegagalan layanan yang ada.

Berdasarkan hasil analisa yang terdapat pada paper yang dirujuk penulis, maka penulis menambahkan fitur-fitur tambahan diluar fitur utama aplikasi sebagai nilai tambah dari aplikasi, yaitu:

1. Fitur popularitas, memungkinkan pengguna untuk mengetahui popularitas dari pengguna lain untuk mengurangi kemungkinan terjadinya *Bad Information*, *Fraud Problem*, dan *Leak of Personal Data*;
2. Fitur *chatting*, memungkinkan untuk bertukar informasi antar pengguna dan meminimalisir terjadinya *Fraud Problem*.

Rincian masalah ketidakpuasan pengguna pada *paper* yang dirujuk terdapat pada Gambar 2.1, sedangkan solusi yang diberikan pengguna terdapat pada Gambar 2.2.

Table 2

Categories of online auction service failures.

Type of service failure	No.	%
<i>Group 1. Service delivery system failures</i>		
Packaging problem	207	23.9
Slow or unavailable	157	18.1
Product defect	142	16.4
Out of stock	61	7.1
Bad information	41	4.7
Alterations and repairs	18	2.1
Hold disaster	14	1.6
Pricing failure	10	1.2
Policy failure	10	1.2
Subtotal, Group 1	660	76.3
<i>Group 2. Buyer needs and requests</i>		
Gap between expectation and perception	83	9.6
Size variation	16	1.8
Special order or request	13	1.5
Admitted buyer error	3	0.3
Subtotal, Group 2	115	13.2
<i>Group 3. Unprompted and unsolicited seller actions</i>		
Seller attention failures	39	4.5
Seller-created embarrassments	22	2.5
Seller fraud problem	21	2.4
Mischarged	8	0.9
Leak of personal data	2	0.2
Subtotal, Group 3	92	10.5
Total	867	100

Gambar 2.1 Hasil Survey Insiden Kegagalan dalam Aplikasi Lelang Online yang Dimuat di *Paper* Rujukan

Table 4

Categories of online auction service recovery strategies.

Recovery strategy	No. (%)	No. of satisfactory recovery	No. of dissatisfactory recovery	Satisfaction with recovery	Retention with recovery
Correction	280 ^a (32.3) ^b	228 ^a (81.4) ^b	52 ^a (18.6) ^b	7.2 ^c (2.4) ^d	6.6 ^c (2.7) ^d
Correction plus	82 (9.5)	77 (93.9)	5 (6.1)	7.7 (2.1)	7.6 (2.5)
Discount	26 (3.0)	23 (88.5)	3 (11.5)	7.0 (2.2)	6.0 (2.9)
Replacement	14 (1.6)	2 (14.3)	12 (85.7)	3.4 (2.7)	3.5 (2.7)
Store credit	16 (1.8)	9 (56.3)	7 (43.7)	6.1 (2.9)	6.4 (2.7)
Apology	54 (6.2)	20 (37.0)	34 (63.0)	3.7 (2.6)	3.4 (2.6)
Refund	67 (7.7)	42 (62.7)	25 (37.3)	6.4 (2.9)	5.2 (3.1)
Unsatisfactory correction	182 (21.0)	0 (0)	182 (100.0)	1.9 (1.3)	1.8 (1.6)
Failure	39 (4.5)	0 (0)	39 (100)	1.3 (0.9)	1.3 (0.9)
escalation	107 (12.4)	0 (0)	107 (100)	1.7 (1.2)	1.7 (1.6)
Nothing					
Total	867 (100)	407 (46.9)	460 (53.1)	4.8 (3.3)	4.4 (3.3)

^a Number.^b %.^c Mean.^d Standard deviation.

Gambar 2.2 Hasil Survey terhadap Fasilitas Perbaikan pada Aplikasi Lelang Online

Penulis juga merujuk pada sebuah *paper* yang membahas mengenai pengaruh popularitas dan reputasi pengguna dari studi kasus pada aplikasi eBay [2]. Kesimpulan dari paper tersebut adalah perilaku pengguna dalam menawar suatu barang juga dipengaruhi oleh popularitas dari pelelang. Jika pelelang memiliki

riwayat yang cukup baik, maka penawar lebih percaya dalam mengajukan penawaran atas barang yang dilelang.

2.1.2 Proses Dasar Pengembangan Perangkat Lunak

Terdapat banyak aspek teknis yang harus diperhatikan untuk mengembangkan aplikasi yang memiliki kualitas baik. Pengembangan aplikasi harus merujuk pada alur proses dalam pengembangan perangkat lunak, sehingga proses pengembangan bisa lebih terkoordinasi dan perubahan dapat dilakukan dengan cepat dan efektif.

Menurut Sommerville dalam bukunya yang berjudul *Software Engineering* edisi ke-9 [3], terdapat empat buah proses fundamental yang harus dilakukan dalam pengembangan perangkat lunak, diantaranya sebagai berikut:

1. *Software specification*
2. *Software design and implementation*
3. *Software validation*
4. *Software evolution*

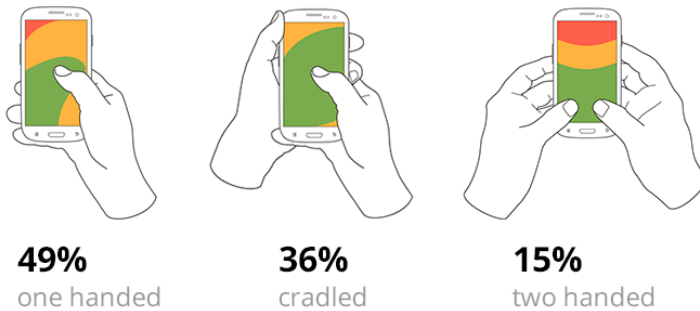
Untuk menghasilkan aplikasi yang berkualitas, efisien, dan siap untuk dikembangkan lebih lanjut, maka setiap poin harus diperhatikan dalam pengembangan aplikasi.

2.1.3 Panduan Menyusun User Experience Aplikasi

Kesuksesan sebuah aplikasi tidak hanya ditentukan dari fungsionalitas fitur dari aplikasi yang sudah dikembangkan, melainkan juga ditentukan dari *user experience* (UX) pada aplikasi tersebut. *User experience* yang baik dapat menyebabkan pengguna merasa nyaman ketika menggunakan aplikasi, dan berpotensi besar untuk menarik pengguna awal aplikasi menjadi pengguna tetap.

Berdasarkan salah satu sumber yang membahas mengenai *user experience* pada UXPlanet [4], terdapat beberapa poin penting yang harus diperhatikan, diantaranya:

1. Satu *screen* untuk satu tugas;
2. Berikan sedikit *space* kosong pada tampilan;



Gambar 2.3 Riset yang Dilakukan oleh Steven Hoober Mengenai Bagaimana Pengguna Memegang Ponsel

3. Navigasi menu yang jelas dan mudah diakses;
4. Dapat dioperasikan dengan mudah menggunakan satu tangan;
5. Konten harus ditayangkan dengan cepat; dan
6. Gunakan *push notification* jika diperlukan.

Dalam pengembangan tugas akhir, selain menggunakan sumber dari UXPlanet yang telah dijabarkan diatas, penulis juga memperhitungkan standar *user experience* pada aplikasi Android yang disarankan oleh Google dalam dokumen ***Mobile App UX Principles*** yang ditulis oleh Stephen Griffiths pada bulan April 2015.

2.1.4 Potensi Pasar dari Sistem Operasi Android

Pasar ponsel pintar yang beroperasi dengan sistem operasi Android adalah yang tertinggi di dunia. Menurut artikel yang dipublikasi oleh *The Verge*, sebanyak 99,6% ponsel pintar yang terjual pada Q4 tahun 2016 mencapai 432 juta ponsel. 352 juta ponsel (81,7%) diantaranya berjalan dengan sistem operasi

Operating System	4Q16 Units	4Q16 Market Share (%)	4Q15 Units	4Q15 Market Share (%)
Android	352,669.9	81.7	325,394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
BlackBerry	207.9	0.0	906.9	0.2
Other OS	530.4	0.1	887.3	0.2
Total	431,539.3	100.0	403,109.4	100.0

Gambar 2.4 Pasar Ponsel Pintar di Dunia pada Q4 2015-2016

Android [5]. Sementara itu, Android menguasai 86.1% pasar ponsel pintar sepanjang masa, diikuti Apple iOS dengan 13.7% [6]. Berdasarkan penjabaran diatas, dapat disimpulkan bahwa pangsa pasar dalam pengembangan aplikasi Android adalah yang terbesar di dunia.

2.2 Analisa Penulis

Pada subbab ini akan dijelaskan analisa yang dilakukan oleh penulis selama pengerjaan tugas akhir.

2.2.1 Analisa terhadap Aplikasi *E-commerce* Umum

Dalam pengerjaan tugas akhir, penulis melakukan analisa dan pengamatan untuk mendapatkan referensi mengenai fitur dan *user experience* yang digunakan oleh aplikasi Android dari *e-commerce* di Indonesia. Aplikasi *e-commerce* yang paling sering dijadikan referensi oleh penulis adalah Tokopedia, Bukalapak dan Carousell.

Terdapat beberapa fitur yang akan dikembangkan pada tugas akhir ini merupakan referensi fitur dari *e-commerce* yang diamati oleh penulis, diantaranya fitur profil, fitur pencarian, fitur ulasan,

dan fitur riwayat. Selain itu, terdapat beberapa unsur *user experience* yang akan dijadikan referensi oleh penulis dalam mengembangkan aplikasi, diantaranya:

1. Satu *screen* untuk satu tugas. Pada aplikasi yang diamati penulis, setiap *activity* yang terdapat pada aplikasi mewakili satu fitur/satu tugas;
2. Memberikan sedikit *space* kosong pada tampilan. Penulis mengamati tampilan aplikasi tidak terlalu *crowded* dan semua informasi penting dapat diperoleh dengan jelas.
3. Navigasi menu yang jelas. Pada aplikasi yang diamati penulis, menu navigasi berada di sisi sebelah kiri atas dari halaman utama dan berisi fitur-fitur utama dari aplikasi. Navigasi *back* untuk kembali ke halaman sebelumnya juga terdapat pada setiap *activity* kecuali *activity* utama, untuk memudahkan pengguna kembali ke laman sebelumnya;
4. Dapat dioperasikan dengan mudah menggunakan satu tangan. Penulis telah mencoba mengoperasikan aplikasi dengan menggunakan satu tangan dan penulis tidak merasa kesulitan saat mengoperasikan aplikasi.
5. Konten harus ditayangkan dengan cepat. Aplikasi *e-commerce* yang diamati oleh penulis telah mengimplementasikan pengiriman *request* ke server API milik mereka pada *background process*, sehingga tidak terjadi *blocking* pada *user interface*.
6. Gunakan *push notification* jika diperlukan. Aplikasi *e-commerce* yang diamati oleh penulis kerap kali menyampaikan notifikasi berupa *follow-up* mengenai status pembayaran dan status barang ketika penulis membeli barang melalui aplikasi tersebut.

Berdasarkan uraian diatas, penulis dapat menyimpulkan bahwa aplikasi *e-commerce* yang diamati telah mengimplementasikan unsur-unsur *user experience* dengan baik.

2.2.2 Analisa terhadap Aplikasi Serupa

Dalam mengembangkan aplikasi, penulis juga melakukan pengamatan terhadap aplikasi-aplikasi lelang *online* yang sudah ada sebelumnya dan masih beroperasi. Aplikasi yang diamati oleh penulis adalah eBay. Dalam analisa yang dilakukan oleh penulis berdasarkan panduan pengguna eBay [7] dan percobaan yang dilakukan penulis, terdapat beberapa batasan dalam alur bisnis pada aplikasi eBay, diantaranya adalah sebagai berikut:

1. Terdapat batasan jumlah *bid* yang dapat diberikan oleh penawar saat mengajukan penawaran;
2. Identitas penawar dirahasiakan;
3. Pada awal penggunaan, penulis kesulitan menemukan barang apa saja yang dilelang, karena dicampur jadi satu dengan barang yang dijual.

Berdasarkan analisa yang telah dilakukan oleh penulis, maka dalam lingkup pengerjaan tugas akhir ini, penulis tidak memberi batasan terhadap jumlah bid yang dapat dilakukan oleh pengguna, dan menampilkan identitas penawar, yaitu nama dari penawar. Hal ini ditujukan untuk memberikan keleluasaan bagi penawar dalam melakukan transaksi lelang. Hal ini dapat berubah jika dalam pengembangan selanjutnya di luar lingkup tugas akhir ditemukan adanya perubahan perilaku pengguna.

Selain itu, penulis juga akan mengembangkan aplikasi dengan alur bisnis yang diadaptasi dari aplikasi *e-commerce* maupun aplikasi serupa yang telah banyak digunakan oleh masyarakat Indonesia, sehingga pengguna dengan cepat merasa familiar saat pertama kali menggunakan aplikasi.

2.2.3 Analisa Potensi Aplikasi

Penulis sempat membaca artikel yang dimuat pada situs *TechInAsia* [8] yang menyatakan bahwa terdapat *trend* baru di Singapura, yaitu lelang *online* melalui fitur *Facebook Live*.

Menurut penulis, hal ini menunjukkan bahwa potensi pasar dari lelang *online* masih memiliki peminat yang cukup tinggi.

2.2.4 Analisa Keamanan Sistem

Keamanan sistem merupakan hal yang sangat vital dalam mengembangkan aplikasi. Oleh sebab itu, penulis memprioritaskan keamanan saat mengembangkan aplikasi. Berdasarkan pengalaman penulis dan sumber yang diperoleh penulis dari forum-forum pemrograman, terdapat beberapa keamanan yang diterapkan oleh penulis saat mengembangkan aplikasi, diantaranya sebagai berikut:

1. HTTPS untuk mengamankan *request* dari aplikasi ke API yang telah dibuat.
2. Memisahkan API menjadi dua kategori, yaitu API privat dan API publik. API privat digunakan untuk transaksi yang bersifat pribadi milik pengguna yang telah melakukan *login* pada aplikasi, sementara API publik dapat diakses oleh semua pengguna, baik yang telah *login* maupun yang tidak *login*.
3. Melakukan *generate token* saat pengguna melakukan *login* pada aplikasi, dan validasi *token* saat pengguna melakukan *request* ke API privat.

2.2.5 Analisa Arsitektur Sistem

Dalam proses mengembangkan aplikasi, penulis memperhitungkan arsitektur sistem yang dibangun, diantaranya penyimpanan data pengguna, baik data yang bersifat transaksional maupun yang bersifat non-transaksional, pemilihan *web-server*, dan pemilihan *server-side scripting*.

1. Nginx sebagai web-server utama
Penulis memilih Nginx atas dasar performa dan kemudahan dalam melakukan *tweaking*. Nginx adalah web-server dengan performa terbaik saat ini, dengan

kemampuan menangani *request* cukup tinggi dan penggunaan memori yang kecil [9].

2. PostgreSQL untuk menyimpan data transaksional
Data-data transaksional seperti data tawaran, data barang, data pengguna disimpan di dalam basis data relasional untuk menjaga integritas data. Dalam hal ini, penulis menggunakan PostgreSQL atas dasar skalabilitas, performa, dan popularitas.
3. MongoDB untuk menyimpan data non-transaksional
Data non-transaksional seperti data *chatting* dan data gambar barang disimpan dalam basis data non-relasional. Penulis mempertimbangkan penggunaan basis data non-relasional dikarenakan untuk memungkinkan terjadinya perubahan penggunaan struktur dan atribut data dengan cepat, sehingga tidak ideal bagi basis data relasional.
4. AmazonAWS S3 untuk menyimpan objek statis
Penulis menggunakan sarana *cloud platform* untuk menyimpan objek statis berupa gambar. Dalam tugas akhir ini, penulis menggunakan AmazonAWS S3 untuk menyimpan objek statis. Pertimbangan penulis dalam untuk menggunakan AWS S3 adalah sebagai berikut:
 1. Skalabilitas
Penulis tidak perlu memikirkan mengenai skalabilitas server dalam penyimpanan data gambar. Semua sudah ditangani oleh Amazon AWS.
 2. Keamanan dan efisiensi
Semua standar keamanan telah ditangani oleh Amazon AWS.

2.3 Lelang

Lelang adalah penjualan barang secara terbuka untuk umum dengan penawaran harga secara tertulis dan lisan yang semakin meningkat atau menurun untuk mencapai harga tertinggi yang didahului dengan pengumuman lelang.

Lelang konvensional biasanya diadakan oleh sebuah perusahaan yang bergerak di bidang jual beli barang lelang dimana proses lelang diadakan di sebuah tempat sehingga konsumen atau penawar bisa berdatangan untuk menjadi peserta dari lelang tersebut. Di Indonesia, terdapat lembaga yang terkait dengan proses lelang, diantaranya Kantor Pelayanan Kekayaan Negara dan Lelang (KPKNL) dan balai lelang baik milik pemerintah maupun swasta[10].

2.4 Online Marketplace

Marketplace (dalam bahasa Indonesia, pasar) adalah tempat bertemunya pembeli dan penjual untuk melakukan sebuah transaksi. *Online marketplace* sendiri merupakan sebuah *platform* untuk mempertemukan pembeli dan penjual secara daring.

Online marketplace merupakan jenis *e-commerce* yang bersifat C2C (*consumer to consumer*), dimana proses transaksi terjadi antara sesama individu yang memiliki akun di *online marketplace* tersebut. Setiap individu dapat bertindak sebagai penjual dan pembeli. Penyedia servis *online marketplace* dalam hal ini hanya bertindak sebagai perantara antar individu[11].

2.5 Java

Java merupakan bahasa pemrograman yang bersifat umum/non spesifik (*general purpose*) dan secara khusus di desain untuk memanfaatkan dependensi implementasi seminimal mungkin. Saat ini Java merupakan bahasa pemrograman yang paling populer dan digunakan secara luas untuk pengembangan aplikasi perangkat lunak berupa *mobile*, *web*, maupun *desktop*.

2.6 JavaScript

JavaScript adalah bahasa pemrograman *client-side* yang *cross-platform* (berjalan di banyak *platform*) dan bersifat bebas untuk dimodifikasi serta diadopsi secara universal[12]. Keunggulan JavaScript, yang juga dikenal dengan nama

ECMAScript, yaitu dapat berjalan di semua *platform* dengan *browser* yang mendukung JavaScript, dan hampir semua *platform* dan *browser* saat ini mendukung JavaScript. Contoh dari aplikasi yang dibangun dengan JavaScript adalah Google Maps yang dapat berjalan di atas Linux, Windows, dan Mac OS. JavaScript juga semakin dan terus berkembang, seperti pertumbuhan pustaka (*library*) yang memudahkan untuk menavigasi dokumen, memilih elemen DOM (*Document Object Model*), membuat animasi, menangani event dan mengembangkan aplikasi Ajax.

JavaScript mulai dikembangkan pada tahun 1995 oleh Brendan Eich saat masih bekerja pada Netscape Communication Corporation. Sampai saat ini JavaScript sudah digunakan untuk membuat beberapa bahasa baru, seperti Objective-J, CoffeeScript, Google Web Toolkit, Scala, Processing.js, dan TypeScript.

2.7 JavaScript Object Notation

JavaScript Object Notation atau yang biasa disebut JSON adalah sebuah format pertukaran data yang berukuran ringan. JSON mudah untuk ditulis dan dimengerti oleh manusia, dan mudah bagi mesin untuk mengurai dan menciptakan. JSON berdasar dari himpunan bagian bahasa pemrograman JavaScript standar ECMA-262 edisi ke-3, Desember 1999[13].

2.8 Android

Android merupakan sistem operasi perangkat bergerak yang berbasis Linux Kernel dan saat ini sedang dikembangkan oleh Google [14]. Android menggunakan antarmuka pengguna yang berbasis manipulasi langsung, dan didesain terutama untuk digunakan pada perangkat bergerak dengan layar sentuh. Android adalah sistem operasi *open source*, dan dirilis di bawah Lisensi Apache. Kode *open source* dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki komunitas pengembang aplikasi dengan jumlah besar yang

memperluas fungsionalitas perangkat, umumnya ditulis dalam versi bahasa pemrograman Java. Hingga Juni 2016, Android telah mencapai versi 6.0 atau *Marshmallow*.

2.9 Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) berbasis IntelliJ IDEA yang dirilis oleh Google dan diperuntukkan untuk pengembang-pengembang yang mengembangkan aplikasi Android [15].

Beberapa fitur yang menjadi andalan Android Studio adalah:

- *Gradle build system* yang fleksibel;
- Emulator yang kaya fitur;
- *Instant Run*, yang mana Android Studio dapat langsung memperbaharui aplikasi tanpa perlu melakukan instalasi aplikasi kembali; dan
- Built-in support untuk *Google platform*.

2.10 Structured Query Language (SQL)

SQL adalah sebuah bahasa pemrograman yang digunakan untuk mengelola data di dalam basis data relasional. SQL pertama kali dikembangkan oleh Donald D. Chamberlain dan Raymond F. Boyce di awal tahun 1970-an. SQL dirilis secara komersil untuk pertama kali oleh Relational Software (sekarang, Oracle Corporation) yang diberi nama OracleV2 [16].

2.11 NoSQL

NoSQL (*not only SQL*) adalah basis data non-relasional yang terdistribusi dan memungkinkan untuk melakukan organisasi dan analisa data secara cepat dengan volume yang besar [17]. Basis data NoSQL biasanya digunakan untuk *cloud computing* dan *big data database*. Terdapat banyak jenis basis data open-source berbasis NoSQL seperti MongoDB, Apache Cassandra, dan Redis.

2.12 PostgreSQL

PostgreSQL (populer dengan nama POSTGRES) adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut perjanjian lisensi BSD [18]. Piranti lunak ini merupakan salah satu basis data yang paling sering digunakan saat ini selain MySQL dan OracleDB. Beberapa fitur yang disediakan oleh PostgreSQL antara lain *indexing*, *trigger*, *foreign key*, *subselect*, dan fitur-fitur basis data relasional yang lain. PostgreSQL dikembangkan oleh sebuah tim dari *University of California, Berkeley*. PostgreSQL dibuat dengan menggunakan bahasa C. Hingga saat ini, PostgreSQL sudah berusia 19 tahun dan masih dikembangkan secara aktif.

2.13 Amazon AWS S3

Amazon AWS S3 adalah sebuah servis penyimpanan yang disediakan oleh Amazon dan digunakan untuk menyimpan data bersifat objek dan memungkinkan untuk mengakses objek tersebut dimanapun. Amazon AWS S3 sudah dilengkapi dengan *web service* yang memudahkan pengembang untuk berinteraksi dengan S3, seperti menambah objek baru, mengubah objek, maupun menghapus objek. Amazon AWS S3 tercatat sudah menyimpan lebih dari satu triliun objek [19].

2.14 ElasticSearch

ElasticSearch adalah sebuah basis data *open-source* yang dikembangkan oleh Elastic dan digunakan untuk membangun sebuah mesin pencarian. ElasticSearch dibangun dengan dasar Apache Lucene yang juga merupakan basis data khusus mesin pencarian. ElasticSearch memiliki antarmuka bersifat RESTful dan dapat menyimpan data dengan cara melakukan *indexing* di cluster yang sudah dibuat. ElasticSearch membagi data pada setiap *shards* sesuai konfigurasi yang telah dibuat sehingga proses temu kembali data dapat dilayani dengan cepat [20], [21]. ElasticSearch telah banyak digunakan oleh banyak perusahaan teknologi, diantaranya eBay, Facebook, dan Netflix [22].

2.15 MongoDB

MongoDB adalah basis data NoSQL yang berorientasi dokumen. MongoDB menggunakan dokumen tipikal JSON (yang disebut BSON) untuk menyimpan data [23]. MongoDB dikembangkan oleh perusahaan perangkat lunak MongoDB Inc. dan dirilis pertama kali pada tahun 2009. MongoDB merupakan basis data NoSQL yang cukup populer saat ini.

(Halaman ini sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan tahap analisis dan perancangan dari aplikasi yang akan dibangun pada Tugas Akhir ini.

3.1 Analisis Sistem

Perangkat lunak yang dibangun adalah aplikasi Lelangapa, yang merupakan aplikasi perangkat bergerak berbasis Android. Aplikasi Lelangapa merupakan aplikasi *marketplace* yang mengusung sistem lelang *online*. Dengan adanya aplikasi Lelangapa, diharapkan pengguna dari semua kalangan yang ingin menjual barang nya dengan cara melelang barang tersebut dapat terbantu. Selain itu, aplikasi Lelangapa yang dikembangkan pada perangkat bergerak berbasis Android, sehingga pengguna dapat mengakses aplikasi di mana saja melalui ponsel pintar Android miliknya.

3.1.1 Analisis Kebutuhan

Berdasarkan pengamatan yang dilakukan pada bab 2.2, terdapat 4 (empat) kebutuhan utama yang harus dipenuhi untuk membuat aplikasi *marketplace* yang familiar bagi pengguna. Empat kebutuhan utama tersebut diantaranya adalah sebagai berikut:

1. Aplikasi memungkinkan pengguna untuk mengelola data diri miliknya;
2. Aplikasi memungkinkan pengguna untuk mengelola barang-barang miliknya;
3. Aplikasi memungkinkan pengguna untuk melakukan aktivitas transaksi utama yang terjadi; dan
4. Aplikasi memungkinkan pengguna untuk menjalankan aktivitas pendukung lain yang mendukung produktivitas pengguna;

Penjabaran lebih lanjut mengenai kebutuhan aplikasi akan dijabarkan pada bab-bab selanjutnya.

3.1.2 Deskripsi Umum Perangkat Lunak

Aplikasi Lelangapa merupakan aplikasi *marketplace* berbasis lelang yang berjalan pada perangkat Android. Aplikasi ini dapat digunakan oleh pengguna umum dari berbagai kalangan untuk melelang barang miliknya dan melakukan penawaran terhadap barang-barang yang dilelang, dengan fitur-fitur sebagai berikut:

1. Pengguna dapat memperbaharui data diri dan profil miliknya.
2. Pengguna sebagai pelemang dapat mengelola barang-barang yang dilelang, meliputi menambah barang baru, merubah informasi barang, dan menghapus barang.
3. Pengguna sebagai pelemang dapat menerima tawaran dari salah satu pengguna untuk dijadikan pemenang lelang, membatalkan tawaran dari salah satu pengguna ketika lelang berlangsung, dan melakukan *block* terhadap salah satu pengguna ketika lelang berlangsung.
4. Pengguna sebagai pelemang dapat membatalkan lelang yang sedang berlangsung.
5. Pengguna sebagai pelemang dapat memberikan review kepada penawar yang memenangkan lelang pada barang yang dilelangnya.
6. Pengguna dapat mencari barang-barang yang dilelang berdasarkan kata kunci dan dapat membatasinya dengan fitur filter.
7. Pengguna sebagai penawar dapat melakukan penawaran kepada suatu barang saat lelang berlangsung.
8. Pengguna sebagai penawar yang memenangkan lelang dapat memberikan review kepada pelemang.
9. Pengguna sebagai penawar dapat melihat riwayat penawaran yang sudah dilakukan.
10. Pengguna sebagai penawar dapat menerima notifikasi jika tawaran nya dikalahkan oleh penawar lainnya.
11. Pengguna dapat saling berkirim pesan singkat dengan pengguna lain.

12. Pengguna dapat melihat daftar ulasan yang diberikan oleh pengguna lain kepada dirinya.

3.1.3 Spesifikasi Kebutuhan Aplikasi

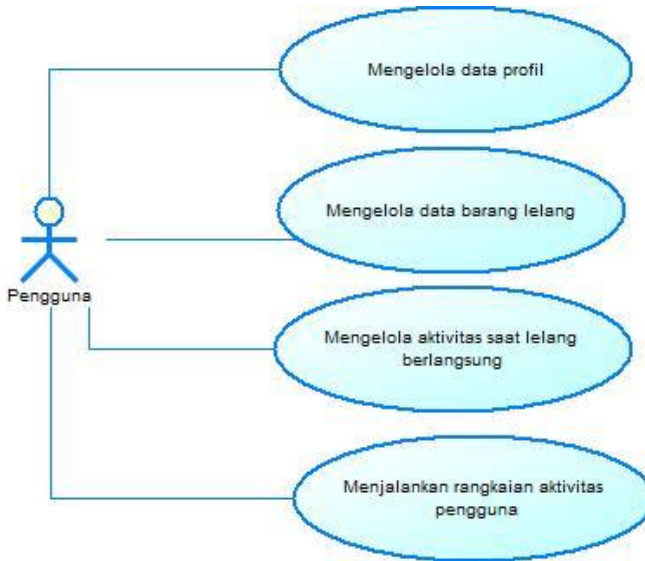
Berdasarkan deskripsi umum sistem pada subbab 3.1.2, dibutuhkan adanya spesifikasi perangkat lunak untuk membangun aplikasi Lelangapa. Spesifikasi kebutuhan akan menjelaskan kebutuhan sistem yang dibutuhkan oleh pengguna.

3.1.3.1 Spesifikasi Kebutuhan Fungsional

Kebutuhan fungsional dari perangkat lunak Lelangapa dapat disimpulkan sebagai berikut:

1. Menangani pengelolaan profil pengguna.
2. Menangani pengelolaan barang-barang lelang.
3. Menangani pengelolaan aktivitas saat lelang berlangsung, meliputi memasukkan penawaran, memilih tawaran sebagai pemenang, membatalkan tawaran dari penawar, melakukan blok pada penawar, dan membatalkan lelang.
4. Menjalankan rangkaian aktivitas pengguna, seperti melakukan pemberian ulasan dan melihat daftar ulasan, melakukan pencarian barang, melihat daftar dan detail riwayat penawaran, menerima notifikasi masuk, ber kirim pesan singkat, dan mengelola barang favorit.

Diagram kasus penggunaan kebutuhan sistem perangkat lunak Lelangapa dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Kasus Penggunaan

3.1.3.2 Spesifikasi Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh perangkat lunak ini berhubungan dengan faktor-faktor berikut:

1. Kebutuhan Performa
Perangkat lunak ini dapat berjalan dalam performa terbaiknya apabila dijalankan pada perangkat keras dengan spesifikasi minimal yang diberikan.
2. Kebutuhan Keamanan
Beberapa fitur dalam perangkat lunak ini hanya dapat diakses oleh pengguna yang sudah terdaftar dengan menggunakan *username* dan kata kunci untuk masuk kedalam sistem.
3. Kebutuhan kualitas perangkat lunak akan dicantumkan pada Tabel 3.1.

Tabel 3.1 Kualitas Perangkat Lunak

No.	Parameter	Deskripsi
1	Ketersediaan	Aplikasi dapat berjalan pada sistem operasi Android, bersifat portabel karena dapat berjalan dimanapun dan kapanpun pada perangkat bergerak.
2	Kenyamanan	Aplikasi dibangun dengan antarmuka pengguna yang mudah dimengerti, enak dilihat, konsisten, mudah dioperasikan, dan tidak membingungkan.

3.1.4 Identifikasi Pengguna

Pengguna adalah entitas-entitas yang terlibat dan berinteraksi secara langsung dengan sistem. Entitas-entitas tersebut dapat berupa manusia maupun sistem ataupun perangkat lunak lain. Pengguna yang didefinisikan untuk perangkat lunak ini adalah pelelang dan penawar.

3.2 Perancangan Sistem

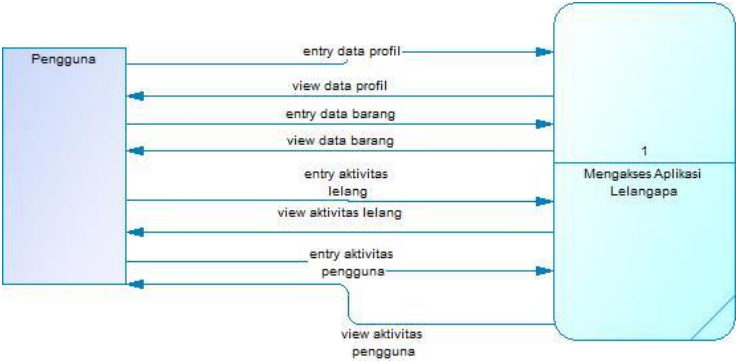
Subbab ini menjelaskan tentang perancangan sistem untuk aplikasi Lelangapa yang dibagi menjadi beberapa subbab, yaitu aliran informasi, deskripsi proses, kamus data, pemodelan data, perancangan antarmuka, perancangan proses menambah, memperbaharui, menghapus, dan mengambil data serta perancangan pembangunan aplikasi pada sistem operasi Android.

Perancangan sistem ini dimaksudkan agar pengembangan aplikasi Lelangapa dibangun secara terstruktur serta memudahkan pengembang dalam mengembangkan aplikasi Lelangapa.

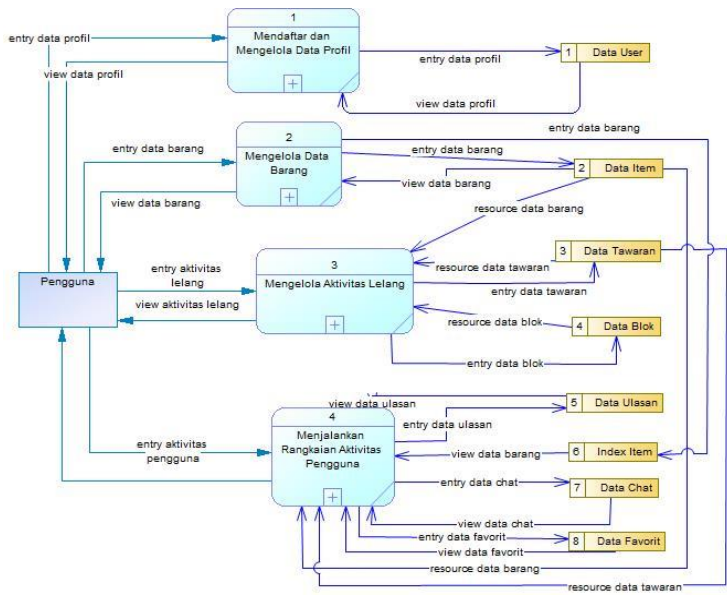
3.2.1 Aliran Informasi

Aliran informasi menjelaskan tentang aliran data yang terjadi pada proses-proses yang ada pada sistem. Rancangan aliran

informasi secara umum untuk perangkat lunak Lelangapa dapat dilihat melalui diagram alur data level 0 pada Gambar 3.2, dan secara lebih terperinci dapat dilihat pada diagram alur data level 1 pada Gambar 3.3.



Gambar 3.2 DFD Level 0



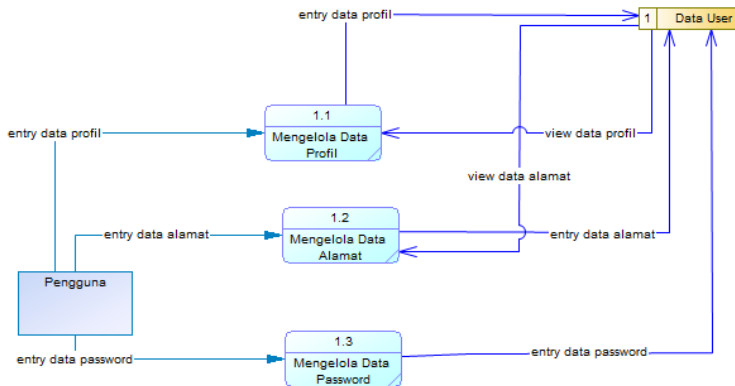
Gambar 3.3 DFD Level 1

3.2.2 Deskripsi Proses

Pada subbab ini akan dijelaskan detail untuk setiap proses yang ada, yaitu proses mendaftar dan mengelola data profil, proses mengelola data barang lelang, proses mengelola aktivitas saat lelang berlangsung, proses memberikan dan menampilkan ulasan, proses menampilkan proses pencarian barang, proses menampilkan daftar dan detail riwayat penawaran, proses menampilkan notifikasi masuk, proses berkirim pesan singkat, dan proses mengelola barang favorit.

3.2.2.1 Proses 1 Mendaftar dan Mengelola Data Profil Pengguna

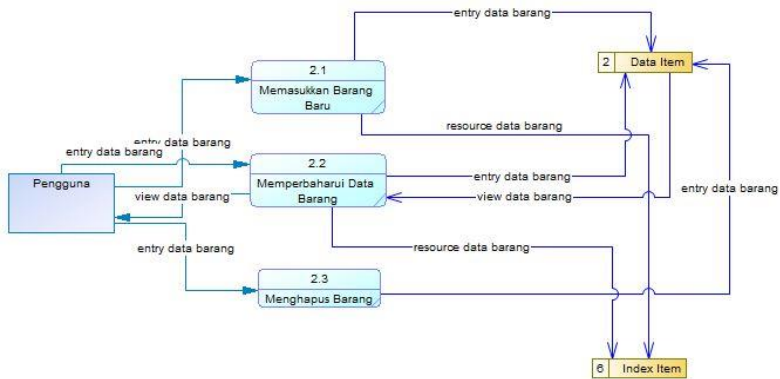
Untuk mengakses perangkat lunak Lelangapa, pengguna tidak diwajibkan untuk mendaftarkan diri terlebih dahulu. Pengguna dapat mengakses fitur-fitur publik pada perangkat lunak Lelangapa. Namun, jika pengguna ingin melakukan transaksi atau melelang barang di Lelangapa, maka pengguna diwajibkan untuk mendaftar / registrasi. Setelah melakukan registrasi, pengguna dapat *login* ke dalam sistem. Pengguna yang telah *login* dapat mengubah dan memperbaharui data profil miliknya dengan mengisi formulir data profil lalu kemudian disimpan ke dalam sistem. Detail proses dapat dilihat pada Gambar 3.4.



Gambar 3.4 Dekomposisi Mengelola Data Profil Pengguna

3.2.2.2 Proses 2 Mengelola Data Barang Lelang

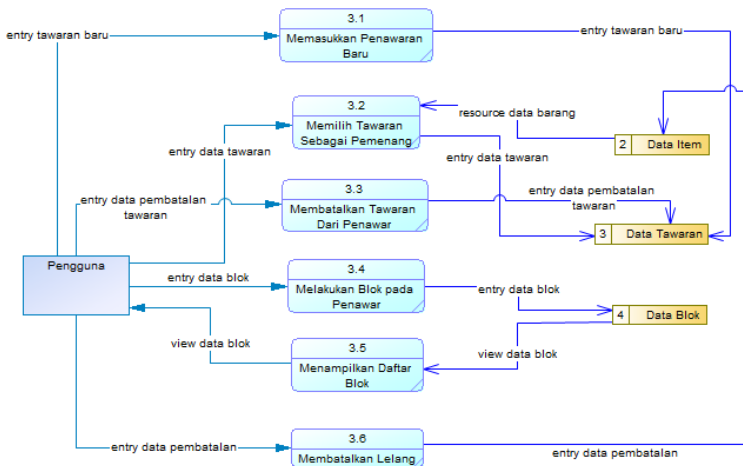
Pengguna yang telah *login* dapat menambahkan data barang baru yang ingin dilelang. Selain itu, pengguna juga dapat memperbaharui data barang yang dilelang, dengan catatan status lelang barang tersebut tidak sedang berlangsung. Jika pengguna memperbaharui data barang saat lelang sudah berakhir, maka barang tersebut akan dilelang kembali. Detail proses dapat dilihat pada Gambar 3.5.



Gambar 3.5 Dekomposisi Mengelola Data Barang Lelang

3.2.2.3 Proses 3 Mengelola Aktivitas saat Lelang Berlangsung

Pengguna yang telah *login* baik penawar maupun pelelang dapat melakukan aktivitas saat lelang sedang berlangsung. Aktivitas yang dapat dilakukan oleh penawar adalah memasukkan penawaran baru. Sementara itu aktivitas yang dapat dilakukan oleh pelelang adalah memilih tawaran sebagai pemenang, membatalkan tawaran dari penawar, melakukan blok pada penawar, dan membatalkan lelang. Detail proses dapat dilihat pada Gambar 3.6. Untuk detail proses masing-masing aktivitas dapat dilihat pada subbab 3.2.2.3.1, 3.2.2.3.2, 3.2.2.3.3, 3.2.2.3.4, 3.2.2.3.5 dan 3.2.2.3.6.



Gambar 3.6 Dekomposisi Mengelola Aktivitas saat Lelang Berlangsung

3.2.2.3.1 Proses 3.1 Memasukkan Penawaran Baru

Pengguna sebagai penawar dapat memasukkan penawaran baru terhadap suatu barang. Penawaran yang diberikan harus lebih tinggi dari penawaran tertinggi saat ini.

3.2.2.3.2 Proses 3.2 Memilih Tawaran Sebagai Pemenang

Pengguna sebagai pelelang dapat memilih salah satu tawaran dari tawaran-tawaran yang sudah diajukan oleh penawar-penawar yang ikut serta dalam aktivitas lelang. Penawar yang tawarannya dipilih oleh pelelang dapat dikatakan sebagai pemenang lelang. Pelelang dapat memilih tawaran saat lelang berlangsung maupun setelah lelang usai. Apabila pelelang memilih pemenang saat lelang masih berlangsung, maka lelang akan langsung dihentikan.

3.2.2.3.3 Proses 3.3 Membatalkan Tawaran dari Penawar

Pengguna sebagai pelelang dapat membatalkan tawaran dari penawar saat lelang berlangsung, jika menurut pelelang tawaran tersebut berpotensi merusak harga atau bersifat *scam*.

3.2.2.3.4 Proses 3.4 Melakukan Blok pada Penawar

Pengguna sebagai pelelang dapat melakukan blok kepada penawar yang berpotensi melakukan pelanggaran atau merusak pasaran lelang. Penawar yang di blok tidak dapat mengajukan penawaran lagi pada sesi lelang tersebut.

3.2.2.3.5 Proses 3.5 Menampilkan Daftar Blok

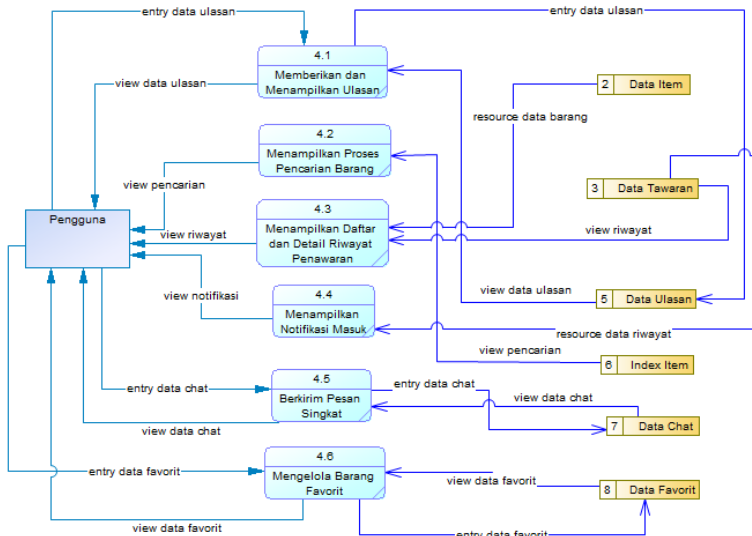
Sistem dapat menampilkan daftar penawar yang diblok pelelang saat sesi lelang sedang berlangsung. Hanya pelelang saja yang dapat melihat daftar blok.

3.2.2.3.6 Proses 3.6 Membatalkan Lelang

Pengguna sebagai pelelang dapat membatalkan lelang yang akan atau sedang berlangsung.

3.2.2.4 Proses 4 Menjalankan Rangkaian Aktivitas Pengguna

Pengguna secara umum baik yang telah melakukan *login* maupun yang belum melakukan *login* dapat menjalankan rangkaian aktivitas. Pengguna yang telah *login* dapat menjalankan rangkaian aktivitas seperti melihat dan memberikan ulasan, melakukan proses pencarian, melihat daftar riwayat penawaran, mendapatkan notifikasi masuk, berkirim pesan singkat, dan mengelola barang favorit. Sementara pengguna yang belum *login* hanya dapat menjalankan aktivitas melakukan pencarian barang saja. Untuk detail dari masing-masing aktivitas dapat dilihat pada sub bab 3.2.2.4.1 hingga 3.2.2.4.6. Detail proses dapat dilihat pada Gambar 3.7.



Gambar 3.7 Dekomposisi Menjalankan Rangkaian Aktivitas Pengguna

3.2.2.4.1 Proses 4.1 Memberikan dan Menampilkan Ulasan

Pengguna sebagai pemenang maupun pelelang dapat memberikan ulasan baru atau mengubah ulasan yang telah dibuat sebelumnya kepada pengguna lain. Pengguna sebagai pemenang dapat memberikan ulasan kepada pelelang, dan sebaliknya pelelang dapat memberikan ulasan kepada pemenang. Pengguna juga dapat melihat ulasan untuk dirinya sebagai pemenang maupun sebagai pelelang.

3.2.2.4.2 Proses 4.2 Menampilkan Proses Pencarian Barang

Pengguna dapat melakukan pencarian barang dengan memasukkan kata kunci pada kolom pencarian, dan sistem akan menampilkan barang-barang yang relevan dengan kata kunci yang dimasukkan.

3.2.2.4.3 Proses 4.3 Menampilkan Daftar dan Detail Riwayat Penawaran

Sistem dapat menampilkan daftar semua barang yang telah ditawarkan oleh pengguna. Selain itu sistem juga dapat menampilkan detail dari masing-masing riwayat.

3.2.2.4.4 Proses 4.4 Menampilkan Notifikasi Masuk

Sistem dapat menampilkan notifikasi masuk kepada pengguna sebagai penawar apabila tawaran miliknya dikalahkan oleh penawar lain nya.

3.2.2.4.5 Proses 4.5 Berkirim Pesan Singkat

Sistem dapat memungkinkan pengguna untuk berkirim pesan singkat dengan pengguna lainnya untuk saling berkomunikasi.

3.2.2.4.6 Proses 4.6 Mengelola Barang Favorit

Sistem dapat menampilkan daftar barang favorit yang dimiliki oleh pengguna dan memungkinkan pengguna memasukkan barang favorit baru maupun membuang barang dari daftar favorit.

3.2.3 Deskripsi Data

Pada subbab ini akan dijelaskan deskripsi data dari entitas-entitas yang terlibat pada sistem sesuai dengan diagram arus data yang telah dijelaskan.

Tabel 3.2 Deskripsi Data (1)

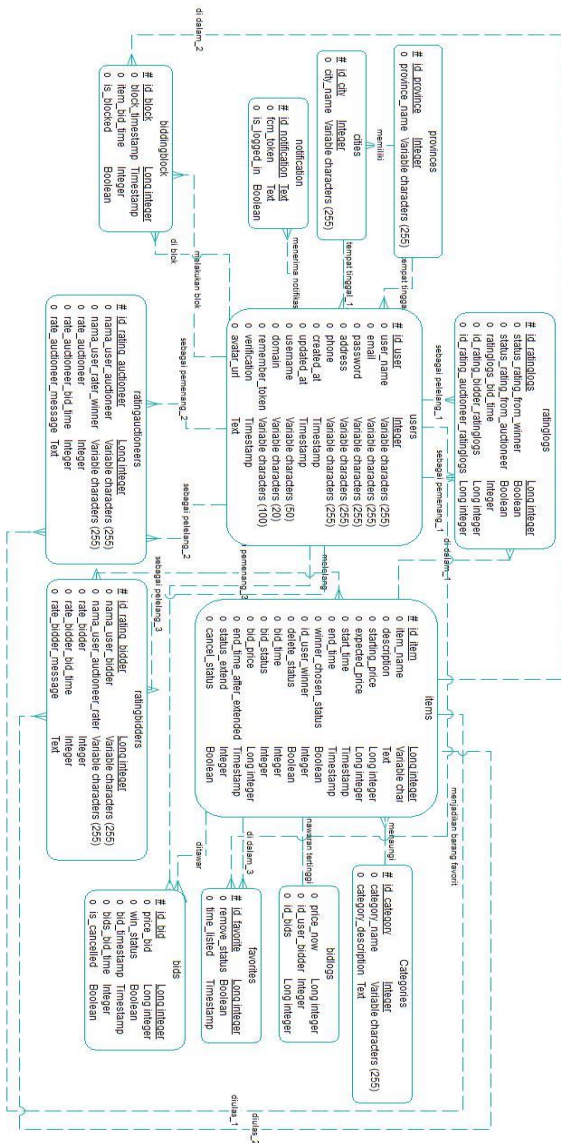
Nama Data	Deskripsi
entry data profil	Data profil yang dikirim oleh pengguna.
view data profil	Data profil yang dikirim oleh server kepada pengguna.
entry data barang	Data barang yang dikirim oleh pengguna.
view data barang	Data barang yang dikirim oleh server ke pengguna.
entry aktivitas lelang	Data aktivitas lelang yang dikirim oleh pengguna.
view aktivitas lelang	Data barang yang dikirim oleh server ke pengguna.
entry data tawaran	Data tawaran yang dikirim oleh pengguna.
entry data blok	Data blok yang dikirim oleh pengguna.
entry aktivitas pengguna	Data aktivitas pengguna yang dikirim oleh pengguna.
entry data ulasan	Data ulasan yang dikirim oleh pengguna.
view data ulasan	Data ulasan yang dikirim oleh server ke pengguna.
entry data <i>chat</i>	Data <i>chat</i> yang dikirim oleh pengguna.
view data <i>chat</i>	Data <i>chat</i> yang dikiriim oleh server ke pengguna.
entry data favorit	Data favorit yang dikirim oleh pengguna.
view data favorit	Data favorit yang dikirim oleh server ke pengguna.
resource data tawaran	Data tawaran yang tersimpan di dalam server yang dibutuhkan oleh suatu proses namun tidak untuk ditampilkan.

Tabel 3.3 Deskripsi Data (2)

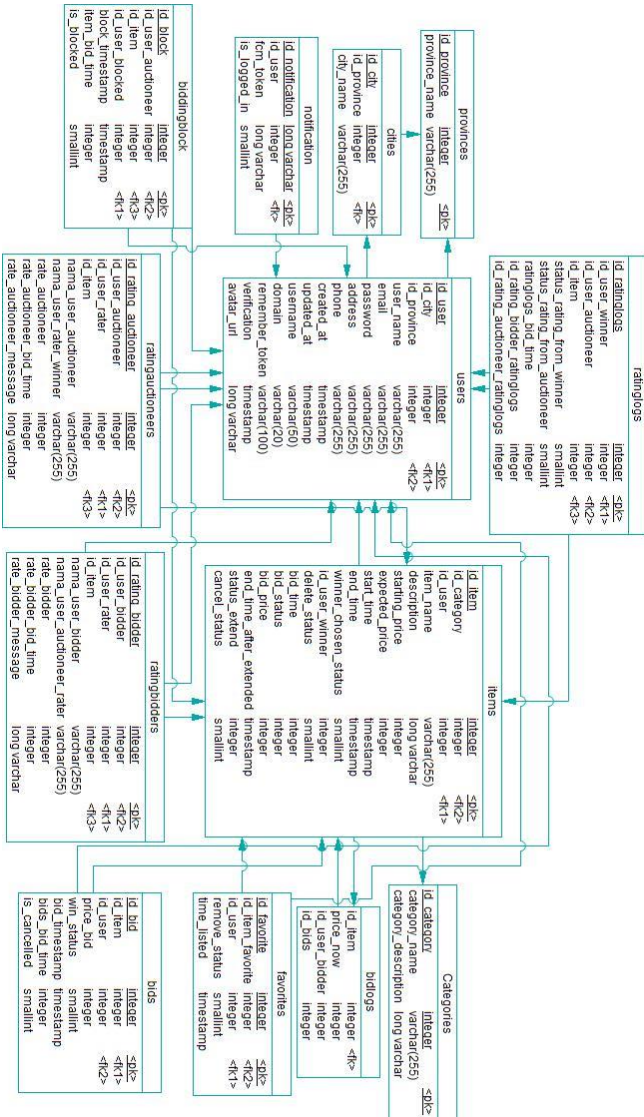
Nama Data	Deskripsi
resource data blok	Data blok pengguna yang tersimpan di dalam server yang dibutuhkan oleh suatu proses namun tidak untuk ditampilkan.
resource data barang	Data barang yang tersimpan di dalam server yang dibutuhkan oleh suatu proses namun tidak untuk ditampilkan.

3.2.4 Pemodelan Data

Pada subbab ini dijelaskan mengenai rancangan model basis data yang digunakan pada perangkat lunak Lelangapa. Rancangan ini disesuaikan dengan kebutuhan sistem yang sudah dibahas pada subbab sebelumnya. *Conceptual Data Model* dan *Physical Data Model* dapat dilihat pada Gambar 3.8 dan Gambar 3.9.



Gambar 3.8 Conceptual Data Model



Gambar 3.9 Physical Data Model

3.2.5 Perancangan Basis Data

Pada subbab ini dijelaskan rancangan basis data yang digunakan pada perangkat lunak Lelangapa. Perangkat lunak Lelangapa menggunakan dua basis data untuk menyimpan data, yaitu PostgreSQL sebagai basis data relasional dan MongoDB sebagai basis data NoSQL.

3.2.5.1 Perancangan Basis Data pada PostgreSQL

Pada subbab ini akan dijelaskan perancangan basis data perangkat lunak Lelangapa pada basis data PostgreSQL. Nama setiap tabel menggunakan bahasa Inggris.

3.2.5.1.1 Tabel *Users*

Tabel ini digunakan untuk menyimpan seluruh data dan informasi dari pengguna. Daftar atribut tabel ini dapat dilihat pada Tabel 3.4.

3.2.5.1.2 Tabel *Items*

Tabel ini digunakan untuk menyimpan seluruh data dan informasi dari semua barang yang dimasukkan pengguna untuk dilelangkan. Daftar atribut tabel ini dapat dilihat pada Tabel 3.5-3.6.

3.2.5.1.3 Tabel *Bids*

Tabel ini digunakan untuk menyimpan semua tawaran yang masuk dari pengguna pada setiap barang yang terdaftar. Daftar atribut tabel ini dapat dilihat pada Tabel 3.7.

3.2.5.1.4 Tabel *Bidlogs*

Tabel ini berfungsi sebagai sebuah tabel pivot. Tabel ini menyimpan data dari tawaran yang memimpin pada masing-masing barang yang dilelang. Daftar atribut tabel ini dapat dilihat pada Tabel 3.8.

3.2.5.1.5 Tabel *BiddingBlock*

Tabel ini digunakan untuk menyimpan data penawar yang di blok oleh pelelang pada suatu sesi lelang. Daftar atribut tabel ini dapat dilihat pada Tabel 3.9.

3.2.5.1.6 Tabel *Categories*

Tabel ini digunakan untuk menyimpan daftar kategori. Daftar atribut tabel ini dapat dilihat pada Tabel 3.10.

3.2.5.1.7 Tabel *Cities*

Tabel ini digunakan untuk menyimpan data kota. Daftar atribut tabel ini dapat dilihat pada Tabel 3.11.

3.2.5.1.8 Tabel *Provinces*

Tabel ini digunakan untuk menyimpan data provinsi. Daftar atribut tabel ini dapat dilihat pada Tabel 3.12.

3.2.5.1.9 Tabel *RatingLogs*

Tabel ini digunakan untuk menyimpan status ulasan pada sebuah transaksi lelang yang sukses. Status yang dimaksudkan disini adalah status pemberian ulasan dari pemenang maupun dari pelelang untuk mengetahui apakah rating sudah diberikan. Daftar atribut tabel ini dapat dilihat pada Tabel 3.13.

3.2.5.1.10 Tabel *RatingAuctioneers*

Tabel ini digunakan untuk menyimpan daftar ulasan yang ditujukan kepada pelelang pada setiap sesi lelang. Daftar atribut tabel ini dapat dilihat pada Tabel 3.14.

3.2.5.1.11 Tabel *RatingBidders*

Tabel ini digunakan untuk menyimpan daftar ulasan yang ditujukan kepada pemenang lelang. Daftar atribut tabel ini dapat dilihat pada Tabel 3.15.

3.2.5.1.12 Tabel *Notification*

Tabel ini digunakan untuk menyimpan token aktif *Firebase Cloud Messaging* pada setiap perangkat keras smartphone yang menginstall perangkat lunak Lelangapa. Daftar atribut tabel ini dapat dilihat pada Tabel 3.16.

3.2.5.1.13 Tabel *Favorites*

Tabel ini digunakan untuk menyimpan seluruh daftar barang favorit dari setiap pengguna. Daftar atribut tabel ini dapat dilihat pada Tabel 3.17.

3.2.5.2 Perancangan Basis Data pada MongoDB

3.2.5.2.1 *Collection ItemImages*

Dokumen ini digunakan untuk menyimpan data dari gambar-gambar barang yang di-*upload* oleh pengguna melalui perangkat lunak Lelangapa. Daftar atribut dokumen ini dapat dilihat pada Tabel 3.18.

3.2.5.2.2 *Collection ChatRoom*

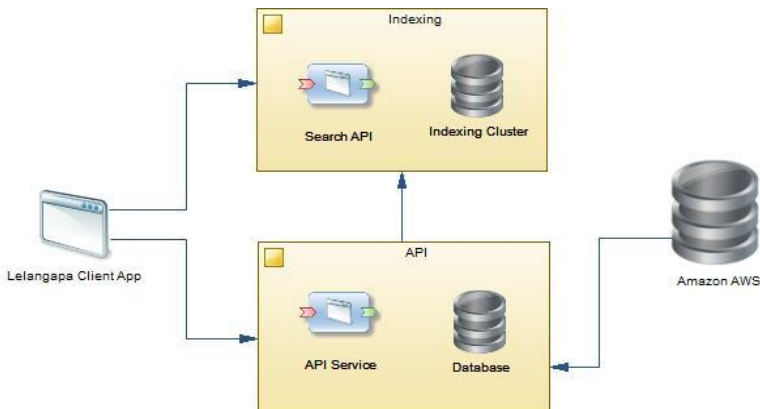
Dokumen ini digunakan untuk menyimpan data dari ruangan chat yang dibentuk oleh pengguna. Setiap pengguna yang memulai interaksi baru pertama kali dengan pengguna lain, data ruangan chat yang terbentuk akan langsung dimasukkan kedalam dokumen ini. Daftar atribut dokumen ini dapat dilihat pada Tabel 3.19.

3.2.5.2.3 Collection UserChat

Dokumen ini digunakan untuk menyimpan data dari percakapan antar pengguna. Data yang disimpan adalah pesan singkat yang dikirimkan oleh pengguna dalam sebuah ruangan *chat*. Daftar atribut dokumen ini dapat dilihat pada Tabel 3.20.

3.2.6 Perancangan Arsitektur Sistem

Pada subbab ini akan dibahas secara terperinci mengenai arsitektur sistem yang digunakan pada perangkat lunak Lelangapa. Terdapat beberapa *layer* pada sistem Lelangapa. Antarmuka pemrograman aplikasi dibangun dengan menggunakan Node.js dan PHP sebagai *server-side scripting*. Untuk menyimpan gambar, Lelangapa menggunakan servis Amazon AWS S3 sebagai penyimpanan / *storage*. Untuk indexing, Lelangapa menggunakan *Elasticsearch*. Server API dan server indexing berada secara terpisah. Hal ini bertujuan untuk mengurangi beban *request* dan beban memori server. Berikut adalah perancangan arsitektur sistem Lelangapa pada Gambar 3.10 berikut.



Gambar 3.10 Perancangan Arsitektur Sistem Lelangapa

3.2.7 Perancangan Antarmuka Pemrograman Aplikasi

Pada subbab ini akan dibahas secara terperinci mengenai rancangan antarmuka pemrograman aplikasi (untuk selanjutnya disebut sebagai API) untuk perangkat lunak Lelangapa. API yang dikembangkan terdiri dari API privat dan API publik.

3.2.7.1 Perancangan API Privat

Pada subbab ini akan dibahas secara terperinci mengenai rancangan API bersifat privat untuk perangkat lunak Lelangapa. Rancangan API ini memiliki kategori akses privat dikarenakan memerlukan token untuk bisa mengaksesnya. Berikut detail dari perancangan modul-modul API akses privat untuk perangkat lunak Lelangapa.

3.2.7.1.1 Perancangan Modul User

Pada perancangan API modul user terdapat empat komponen modul sebagai berikut:

1. Modul Informasi Pengguna
(GET /users/profile/info)

Modul ini digunakan untuk mendapatkan informasi profil pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.21. Hasil *response body* berupa dokumen JSON dengan komposisi data yang dapat dilihat pada Tabel 3.22 dan Tabel 3.23.

2. Modul Ubah Profil Pengguna
(PUT /users/update/profile)

Modul ini digunakan untuk melakukan pembaharuan pada data profil pengguna. Pembaharuan yang ditangani oleh aplikasi ini meliputi perubahan alamat email dan perubahan nomor telepon. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.24. Hasil *response body* berupa dokumen JSON dengan komposisi data yang dapat dilihat pada Tabel 3.25.

3. Modul Ubah Alamat Pengguna
(PUT /users/update/address)

Modul ini digunakan untuk melakukan pembaharuan pada data alamat pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.26. Hasil *response body* berupa dokumen JSON dengan komposisi data yang dapat dilihat pada Tabel 3.27.

4. Modul Ubah Password Pengguna
(PUT /users/update/password)

Modul ini digunakan untuk melakukan pembaharuan pada data *password* pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.28. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.29.

3.2.7.1.2 Perancangan Modul Item

Pada perancangan API modul item terdapat lima komponen modul sebagai berikut:

1. Modul Informasi Barang
(GET /items/:itemId/info)

Modul ini digunakan untuk mendapatkan informasi suatu barang. Informasi ini bersifat privat untuk pelelang barang tersebut. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.30. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.31 dan 3.32.

2. Modul Lelang Barang
(POST /items)

Modul ini digunakan untuk melelangkan barang baru. Klien akan mengirim *request* dengan komposisi data pada Tabel 3.33. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.34.

3. Modul *Edit* Lelang Barang
(PUT /items)

Modul ini digunakan untuk melakukan pembaharuan data barang yang dilelang. Informasi awal dari barang yang akan diperbaharui diperoleh dari modul informasi barang pada poin (1). Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.35. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.36.

4. Modul Hapus Barang
(PUT /items/delete)

Modul ini digunakan untuk melakukan penghapusan suatu barang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.37. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.38.

5. Modul Mendapatkan Daftar Barang Milik Pengguna
(GET /users/items)

Modul ini digunakan untuk mendapatkan daftar barang milik pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.39. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.40 dan 3.41.

3.2.7.1.3 Perancangan Modul *Feedback*

Pada perancangan API modul feedback terdapat sepuluh komponen modul sebagai berikut:

1. Modul Mendapatkan Daftar Ulasan sebagai Pelelang
(GET /users/ratings/as/auctioneer)

Modul ini digunakan untuk mendapatkan daftar ulasan pengguna pribadi sebagai pelelang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.42. Hasil *response body* berupa dokumen JSON

dengan komposisi data dapat dilihat pada Tabel 3.43 dan 3.44.

2. Modul Mendapatkan Daftar Ulasan sebagai Pemenang (GET /users/ratings/as/winner)
Modul ini digunakan untuk mendapatkan daftar ulasan pengguna pribadi sebagai pemenang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.45. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.46 dan 3.47.
3. Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pemenang (GET /users/ratings/for/winner)
Modul ini digunakan untuk mendapatkan daftar ulasan yang dapat diberikan oleh pengguna sebagai pelelang kepada pengguna lain sebagai pemenang lelang atas barang yang dilelangnya. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.48. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.49 dan 3.50.
4. Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang (GET /users/ratings/for/auctioneer)
Modul ini digunakan untuk mendapatkan daftar ulasan yang dapat diberikan oleh pengguna sebagai pemenang lelang kepada pengguna lain sebagai pelelang atas barang yang dimenangkannya. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.51. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.52 dan 3.53.

5. Modul Memasukkan Ulasan Baru untuk Pemenang
(POST /ratings/winner)

Modul ini memungkinkan pengguna sebagai pelelang untuk memberikan ulasan kepada pengguna lain sebagai pemenang lelang atas barang yang dilelangnya. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.54. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.55.

6. Modul Memasukkan Ulasan Baru untuk Pelelang
(POST /ratings/auctioneer)

Modul ini memungkinkan pengguna sebagai pemenang lelang untuk memberikan ulasan kepada pengguna lain sebagai pelelang atas barang yang dimenangkannya. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.56. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.57.

7. Modul Memperbaharui Ulasan untuk Pemenang
(PUT /ratings/winner)

Modul ini memungkinkan pengguna sebagai pelelang untuk memperbaharui ulasan yang telah diberikannya kepada pengguna lain sebagai pemenang lelang atas barang yang dilelangnya. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.58. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.59.

8. Modul Memperbaharui Ulasan untuk Pelelang
(PUT /ratings/auctioneer)

Modul ini memungkinkan pengguna sebagai pelelang untuk memperbaharui ulasan yang telah diberikannya kepada pengguna lain sebagai pemenang lelang atas barang yang dilelangnya. Klien akan mengirimkan

request dengan komposisi data pada Tabel 3.60. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.61.

9. Modul Mendapatkan Detail Ulasan untuk Pemenang (GET /ratings/winner/:ratinglogsId/details)
Modul ini digunakan untuk mendapatkan informasi detail ulasan untuk pemenang yang telah diberikan oleh pengguna sebagai pelelang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.62. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.63 dan 3.64.
10. Modul Mendapatkan Detail Ulasan untuk Pelelang (GET /ratings/auctioneer/:ratinglogsId/details)
Modul ini digunakan untuk mendapatkan informasi detail ulasan untuk pelelang yang telah diberikan oleh pengguna sebagai pemenang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.65. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.66 dan 3.67.

3.2.7.1.4 Perancangan Modul Riwayat Penawaran

Pada perancangan API modul riwayat penawaran terdapat dua komponen modul sebagai berikut:

1. Modul Mendapatkan Seluruh Daftar Riwayat Penawaran (GET /users/:userId/bids/history)
Modul ini digunakan untuk mendapatkan semua daftar barang-barang yang telah ditawarkan oleh pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.68. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.69 dan 3.70.

2. Modul Mendapatkan Detail suatu Riwayat Penawaran (GET /users/:userId/bids/history/list/:itemId:bidTime)
Modul ini digunakan untuk mendapatkan semua daftar riwayat harga penawaran yang diberikan oleh pengguna untuk suatu barang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.71. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.72 dan 3.73.

3.2.7.1.5 Perancangan Modul Favorit

Pada perancangan API modul favorit terdapat empat komponen modul sebagai berikut:

1. Modul Mendapatkan Status Favorit suatu Barang (GET /favorites/:userId:itemId)
Modul ini digunakan untuk mendapatkan informasi mengenai status favorit suatu barang terhadap pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.74. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.75 dan 3.76.
2. Modul Mendapatkan Daftar Barang Favorit (GET /favorites/:userId)
Modul ini digunakan untuk mendapatkan seluruh daftar barang favorit pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.77. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.78 dan 3.79.
3. Modul Memasukkan Barang sebagai Favorit (POST /favorites)
Modul ini digunakan untuk memasukkan suatu barang ke dalam daftar barang favorit pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel

3.80. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.81.

4. Modul Menghapus Barang dari Favorit
(PUT /favorites)

Modul ini digunakan untuk menghapus suatu barang dari daftar barang favorit pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.82. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.83.

3.2.7.1.6 Perancangan Modul Pesan Singkat (*Chat*)

Pada perancangan API modul pesan singkat (*chat*) terdapat dua komponen modul sebagai berikut:

1. Modul Mendapatkan Daftar Ruang *Chat*
(GET /chat/rooms)

Modul ini digunakan untuk mendapatkan daftar ruangan *chat* pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.84. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.85 dan 3.86.

3.2.7.1.7 Perancangan Modul Daftar Tawaran

Pada perancangan API modul daftar tawaran terdapat satu komponen modul sebagai berikut:

1. Modul Mendapatkan Daftar Tawaran suatu Barang
(GET /bids/item/:itemId/offers)

Modul ini digunakan untuk mendapatkan daftar tawaran pada suatu barang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.87. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.88 dan 3.89.

3.2.7.1.8 Perancangan Modul Notifikasi

Pada perancangan API modul notifikasi terdapat dua komponen modul sebagai berikut:

1. Modul Memasukkan Token FCM
(POST /fcm/token/submit)
Modul ini digunakan untuk memasukkan token FCM yang telah didapatkan kedalam basis data di server. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.90. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.91.

2. Modul Mengubah Status *Login* ketika *Logout*
(POST /fcm/token/logout)
Modul ini digunakan untuk mengubah status *login* pengguna ketika pengguna melakukan *logout*, sehingga notifikasi tidak lagi dikirimkan ke pengguna yang telah melakukan *logout*. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.92. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.93.

3.2.7.1.9 Perancangan Modul Blok

Pada perancangan API modul blok terdapat tiga komponen modul sebagai berikut:

1. Modul Memblok Pengguna
(POST /bids/block)
Modul ini digunakan oleh pengguna sebagai pelelang untuk melakukan blok terhadap pengguna lain sebagai penawar pada suatu sesi lelang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.94. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.95.

2. Modul Membebaskan Pengguna
(POST /bids/unblock)

Modul ini digunakan oleh pengguna sebagai pelelang untuk membebaskan pengguna lain sebagai penawar yang telah di blok. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.96. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.97.

3. Modul Mendapatkan Daftar Blok
(GET /bids/block/item/:itemId/bidtime/:bidTime/list)

Modul ini digunakan untuk mendapatkan daftar penawar yang telah di blok oleh pelelang dalam suatu sesi lelang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.98. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.99 dan 3.100.

3.2.7.1.10 Perancangan Modul Pilih Pemenang

Pada perancangan API modul pilih pemenang terdapat satu komponen modul sebagai berikut:

1. Modul Memilih Pemenang Lelang
(POST /bids/select/winner)

Modul ini digunakan untuk memilih pemenang lelang setelah waktu lelang telah usai. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.101. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.102.

3.2.7.1.11 Perancangan Modul Validasi Token

Pada perancangan API modul validasi token terdapat satu komponen modul sebagai berikut:

1. Modul Melakukan Validasi Token
(GET /validation/token)

Modul ini digunakan untuk melakukan validasi token saat pengguna yang telah melakukan *login* membuka perangkat lunak Lelangapa. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.103. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.104.

3.2.7.2 Perancangan API Publik

Pada subbab ini akan dibahas secara terperinci mengenai rancangan antarmuka pemrograman aplikasi bersifat publik untuk perangkat lunak Lelangapa. API yang bersifat publik tidak perlu menyertakan token untuk mengaksesnya. Berikut detail perancangan API akses publik untuk perangkat lunak Lelangapa.

3.2.7.2.1 Modul Item

Pada perancangan API modul item terdapat satu komponen modul sebagai berikut:

1. Modul Mendapatkan Daftar Barang
(GET /public/users/:userId/items)

Modul ini digunakan untuk mendapatkan daftar barang yang dilelang milik pengguna lain. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.105. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.106 dan 3.107.

3.2.7.2.2 Modul User

Pada perancangan API modul user terdapat tiga komponen modul sebagai berikut:

1. Modul *Login*
(POST /public/users/login)

Modul ini digunakan untuk melakukan *login* akun pengguna. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.108. Hasil *response body*

berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.109 dan 3.110.

2. Modul *Register*
(POST /public/users/register)
Modul ini digunakan untuk melakukan registrasi akun baru. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.111. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.112.
3. Modul Melakukan Validasi Input *Username*
(GET /public/checker/username/:username)
Modul ini digunakan untuk memeriksa apakah *username* yang akan dibuat sudah ada yang menggunakan. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.113. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.114.
4. Modul Melakukan Validasi Input *Domain*
(GET /public/checker/domain/:domain)
Modul ini digunakan untuk memeriksa apakah domain yang akan dibuat sudah ada yang menggunakan. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.115. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.116.

3.2.7.2.3 Modul *Feedback*

Pada perancangan API modul user terdapat empat komponen modul sebagai berikut:

1. Modul Mendapatkan Daftar Ulasan sebagai Pemenang
(GET /public/users/:userId/ratings/as/winner)

Modul ini digunakan untuk mendapatkan daftar ulasan sebagai pemenang pada pengguna lain. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.117. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.118 dan 3.119.

2. Modul Mendapatkan Daftar Ulasan sebagai Pelelang
(GET /public/users/:userId/ratings/as/auctioneer)
Modul ini digunakan untuk mendapatkan daftar ulasan sebagai pelelang pada pengguna lain. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.120. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.121 dan 3.122.
3. Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang
(GET /public/users/:userId/ratings/as/winner/average)
Modul ini digunakan untuk mendapatkan rata-rata ulasan sebagai pemenang pada pengguna lain. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.123. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.124.
4. Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang
(GET /public/users/:userId/ratings/as/auctioneer/average)
Modul ini digunakan untuk mendapatkan rata-rata ulasan sebagai pelelang pada pengguna lain. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.125. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.126.

3.2.7.2.4 Modul Riwayat

Pada perancangan API modul user terdapat satu komponen modul sebagai berikut:

1. Modul Mendapatkan Daftar Riwayat Penawaran
(GET /public/users/:userId/bids/history)
Modul ini digunakan untuk mendapatkan daftar riwayat penawaran yang dilakukan pengguna lain. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.127. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.128 dan 3.129.

3.2.7.2.5 Modul Detail Item

Pada perancangan API modul user terdapat dua komponen modul sebagai berikut:

1. Modul Mendapatkan Detail Barang
(GET /public/items/:itemId/details)
Modul ini digunakan untuk mendapatkan informasi detail dari suatu barang yang dilelang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.130. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.131 dan 3.132.
2. Modul Mendapatkan Peringkat Penawaran pada Detail Barang
(GET /public/items/:itemId/details/ranks)
Modul ini digunakan untuk mendapatkan informasi detail peringkat penawaran suatu barang yang dilelang. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.133. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.134 dan 3.135.

3.2.7.2.6 Modul Pencarian

Pada perancangan API modul user terdapat tiga komponen modul sebagai berikut:

1. Modul Pencarian berdasarkan Kata Kunci
(POST /apis/search)
Modul ini digunakan untuk melakukan pencarian berdasarkan kata kunci yang dimasukkan. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.136. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.139-141.
2. Modul Pencarian berdasarkan Penyaringan
(POST /apis/search/filter)
Modul ini digunakan untuk melakukan pencarian berdasarkan kata kunci dan parameter penyaringan. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.137. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.139-141.
3. Modul Menampilkan semua Barang pada suatu Kategori
(POST /apis/search/category)
Modul ini digunakan untuk menampilkan seluruh barang yang merupakan bagian dari sebuah kategori. Klien akan mengirimkan *request* dengan komposisi data pada Tabel 3.138. Hasil *response body* berupa dokumen JSON dengan komposisi data dapat dilihat pada Tabel 3.139-141.

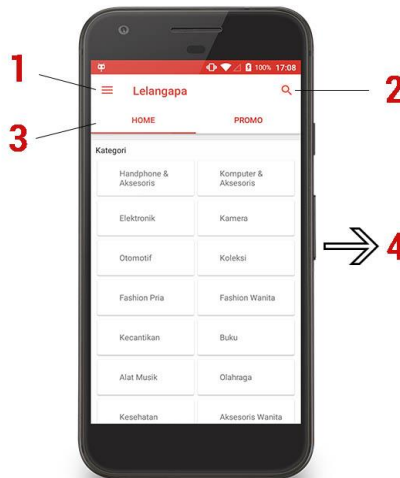
3.2.8 Perancangan Antarmuka Perangkat Lunak

Pada subbab ini akan dibahas secara terperinci mengenai rancangan antarmuka pada perangkat lunak Lelangapa.

3.2.8.1 Antarmuka Halaman Utama

Gambar 3.11 merupakan rancangan halaman utama. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.11.

1. *Navigation Drawer* untuk opsi menu;
2. Tombol *search* untuk melakukan pencarian;
3. *TabLayout* untuk memilih laman yang ada di halaman utama. Untuk saat ini, laman “Promo” tidak digunakan; dan
4. *RecyclerView* berisi daftar kategori.

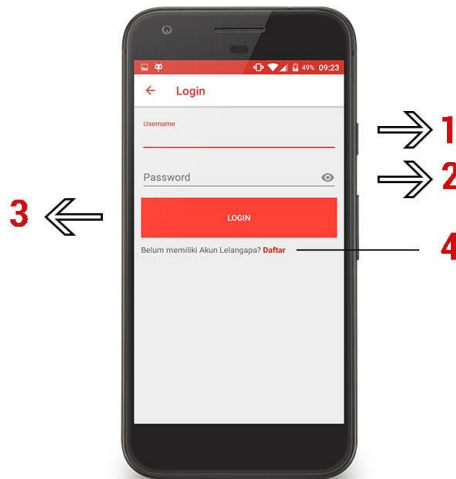


Gambar 3.11 Rancangan Antarmuka Halaman Utama

3.2.8.2 Antarmuka Halaman *Login*

Gambar 3.12 merupakan rancangan halaman *login*. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.12.

1. *EditText* untuk memasukkan *username*;
2. *EditText* untuk memasukkan *password* atau kata kunci;
3. *Button* untuk melakukan *login*; dan
4. *Clickable TextView* untuk mendaftar sebagai pengguna baru.

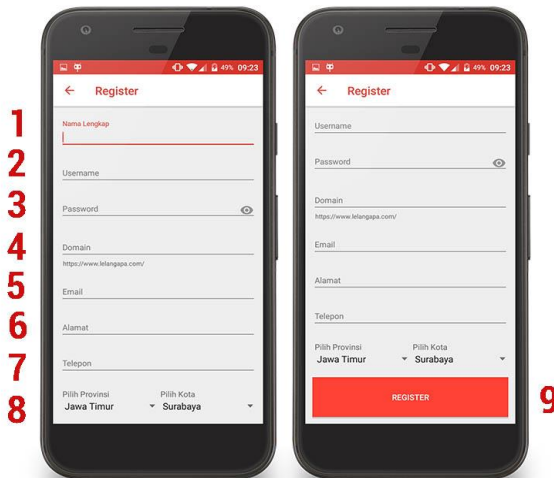


Gambar 3.12 Rancangan Antarmuka Halaman *Login*

3.2.8.3 Antarmuka Halaman Register

Gambar 3.13 merupakan rancangan halaman *register*. Halaman *register* digunakan untuk melakukan proses pendaftaran pengguna baru. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.13:

1. *EditText* untuk memasukkan nama lengkap;
2. *EditText* untuk memasukkan *username*
3. *EditText* untuk memasukkan *password*
4. *EditText* untuk memasukkan domain
5. *EditText* untuk memasukkan email
6. *EditText* untuk memasukkan alamat
7. *EditText* untuk memasukkan nomor telepon
8. *Spinner* untuk memilih provinsi dan kota
9. *Button* untuk melakukan registrasi.

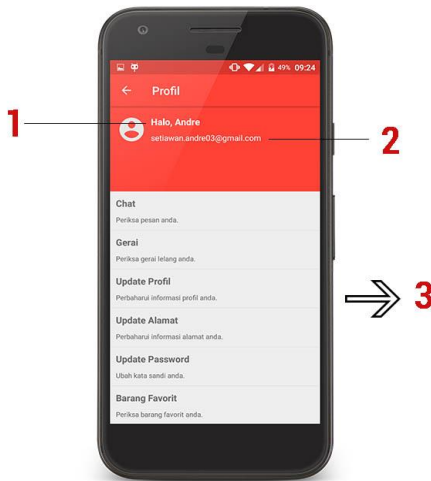


Gambar 3.13 Rancangan Antarmuka Halaman Register

3.2.8.4 Antarmuka Halaman Profil

Gambar 3.14 merupakan rancangan halaman profil. Halaman profil berisi informasi mengenai pengguna yang telah *login* dan daftar menu yang dapat diakses. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.14.

1. *TextView* untuk informasi nama
2. *TextView* untuk informasi email
3. *RecyclerView* berisi daftar menu yang dapat diakses

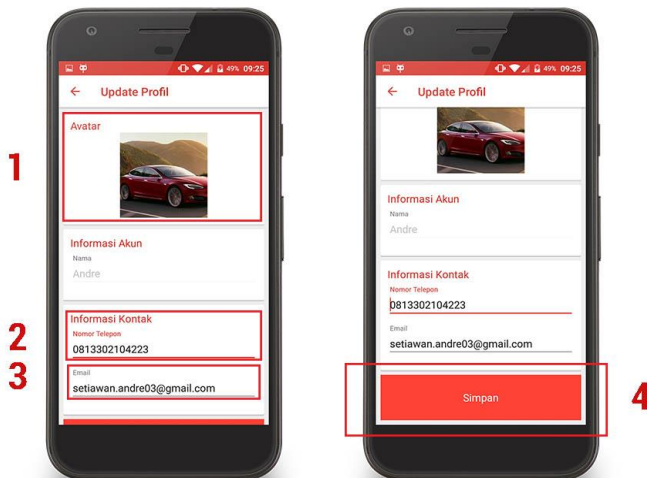


Gambar 3.14 Rancangan Antarmuka Halaman Profil

3.2.8.5 Antarmuka Halaman *Edit Profil*

Gambar 3.15 merupakan rancangan halaman edit profil. Halaman *Edit Profil* digunakan untuk memperbaharui data pengguna yang telah *login*. Beberapa data yang dapat diperbaharui antara lain avatar, alamat email, dan nomor telepon. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.15:

1. *ImageView* avatar
2. *EditText* berisi data alamat email
3. *EditText* berisi data nomor telepon
4. *Button* untuk melakukan edit profil



Gambar 3.15 Rancangan Antarmuka Halaman *Edit Profil*

3.2.8.6 Antarmuka Halaman *Edit Alamat*

Gambar 3.16 merupakan rancangan halaman edit alamat. Halaman edit alamat digunakan untuk melakukan perubahan detail alamat dari pengguna yang telah melakukan *login* sebelumnya. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.16:

1. *EditText* berisi data alamat.
2. *Button* untuk melakukan edit alamat.

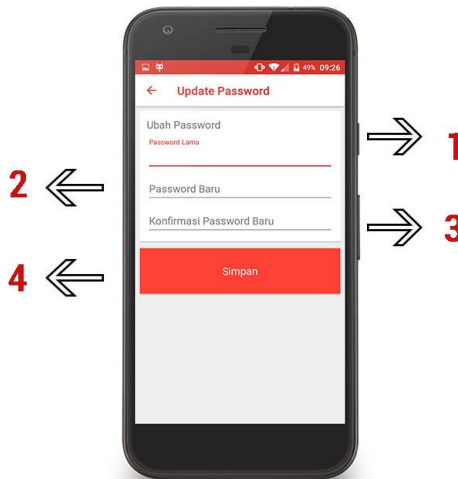


Gambar 3.16 Rancangan Antarmuka Halaman *Edit Alamat*

3.2.8.7 Antarmuka Halaman Ubah Password

Gambar 3.17 merupakan rancangan halaman ubah *password*. Halaman ubah *password* digunakan untuk mengubah *password* atau kata kunci dari pengguna yang telah *login* sebelumnya. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.17.

1. *EditText* untuk memasukkan *password* lama
2. *EditText* untuk memasukkan *password* baru
3. *EditText* untuk memasukkan konfirmasi *password* baru
4. *Button* untuk mengirim data ke server.

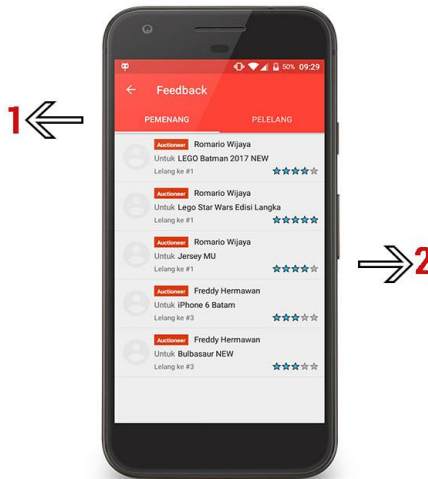


Gambar 3.17 Rancangan Antarmuka Halaman Ubah Password

3.2.8.8 Antarmuka Halaman Beri *Feedback*

Gambar 3.18 merupakan rancangan halaman beri *feedback*. Halaman beri *feedback* digunakan untuk memberikan daftar *feedback* atau ulasan yang dapat diberikan kepada pengguna lain. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.18.

1. *TabLayout* untuk memilih laman sebagai pemenang atau sebagai pelelang;
2. *RecyclerView* untuk menampilkan daftar ulasan yang dapat diberikan

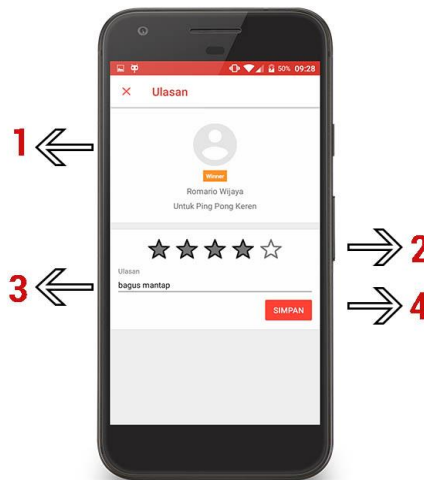


Gambar 3.18 Rancangan Antarmuka Halaman Beri *Feedback*

3.2.8.9 Antarmuka Halaman Detail Beri *Feedback*

Gambar 3.19 merupakan rancangan halaman detail beri *feedback*. Halaman detail beri *feedback* digunakan untuk memberikan *feedback* atau ulasan kepada pengguna lain. Berikut beberapa penjelasan masing-masing komponen yang terdapat pada Gambar 3.19.

1. *ImageView* untuk menampilkan avatar pengguna;
2. *RatingBar* untuk memberikan ulasan berupa peringkat;
3. *EditText* untuk menuliskan ulasan.
4. *Button* untuk menyimpan ulasan.

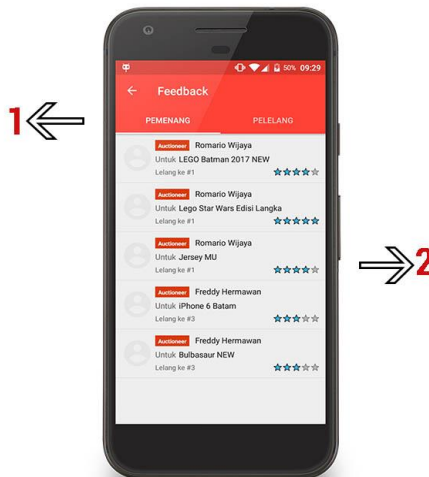


Gambar 3.19 Rancangan Antarmuka Halaman Detail Beri *Feedback*

3.2.8.10 Antarmuka Halaman *Feedback* Anda

Gambar 3.20 merupakan rancangan halaman feedback anda. Halaman feedback anda digunakan untuk menampilkan seluruh feedback atau ulasan yang diberikan oleh pengguna lain. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.20.

1. *TabLayout* untuk memilih laman sebagai pemenang atau sebagai pelelang;
2. *RecyclerView* untuk menampilkan daftar ulasan yang dapat diberikan

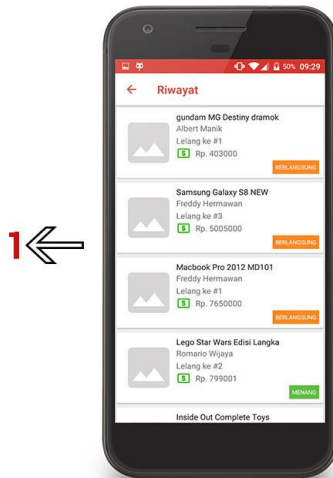


Gambar 3.20 Rancangan Antarmuka Halaman *Feedback* Anda

3.2.8.11 Antarmuka Halaman Riwayat

Gambar 3.21 merupakan rancangan halaman riwayat. Halaman riwayat digunakan untuk menampilkan seluruh riwayat penawaran yang dilakukan oleh pengguna. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.21.

1. *RecyclerView* untuk menampilkan daftar riwayat penawaran.



Gambar 3.21 Rancangan Antarmuka Halaman Riwayat

3.2.8.12 Antarmuka Halaman Detail Riwayat

Gambar 3.22 merupakan rancangan halaman detail riwayat. Halaman detail riwayat digunakan untuk menampilkan informasi lanjutan mengenai riwayat sebuah penawaran. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.22.

1. *ImageView* untuk menampilkan gambar barang;
2. *TextView* untuk menampilkan nama barang;
3. *TextView* untuk menampilkan nama pelemang;
4. *TextView* untuk menampilkan sesi lelang;
5. *RecyclerView* untuk menampilkan waktu penawaran yang dilakukan beserta harga tawaran.

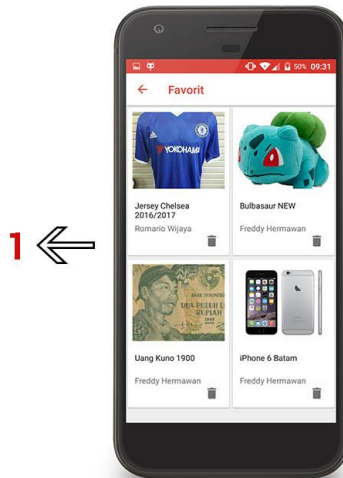


Gambar 3.22 Rancangan Antarmuka Halaman Detail Riwayat

3.2.8.13 Antarmuka Halaman Favorit

Gambar 3.23 merupakan rancangan halaman favorit. Halaman favorit digunakan untuk menyimpan seluruh barang favorit yang telah dipilih pengguna. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.23.

1. *RecyclerView* untuk menampilkan daftar barang-barang favorit.

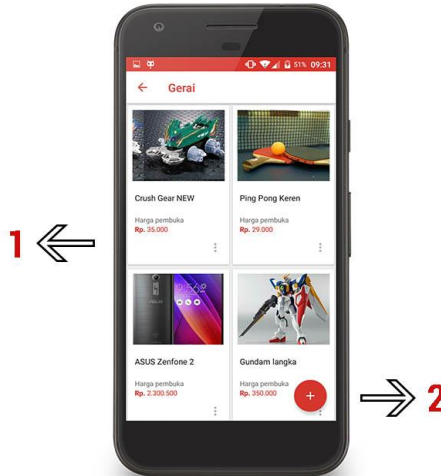


Gambar 3.23 Rancangan Antarmuka Halaman Favorit

3.2.8.14 Antarmuka Halaman Gerai

Gambar 3.24 merupakan rancangan halaman gerai. Halaman gerai berisi barang-barang yang dilelang oleh pengguna. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.24.

1. *RecyclerView* untuk menampilkan seluruh daftar barang yang dilelang oleh pengguna dan juga menu untuk menghapus barang;
2. *FloatingActionButton* yang menuju ke halaman lelang barang.

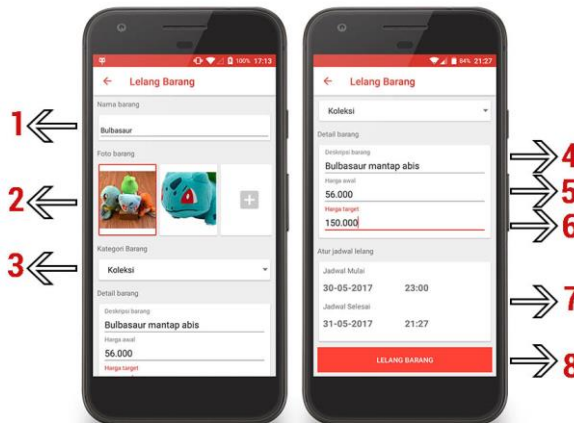


Gambar 3.24 Rancangan Antarmuka Halaman Gerai

3.2.8.15 Antarmuka Halaman Lelang Barang

Gambar 3.25 merupakan rancangan halaman lelang barang. Halaman lelang barang digunakan untuk melelang barang baru. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.25.

1. *EditText* untuk memasukkan nama barang;
2. *RecyclerView* yang berisi *CardView* untuk memasukkan dan menampilkan gambar-gambar barang yang dimasukkan pengguna;
3. *Spinner* untuk menampilkan daftar kategori;
4. *EditText* untuk memasukkan deskripsi barang;
5. *EditText* untuk memasukkan harga pembuka;
6. *EditText* untuk memasukkan harga ekspektasi;
7. *Clickable TextView* untuk memasukkan waktu mulai dan waktu selesai lelang; dan
8. Button untuk mengirim data barang ke server.

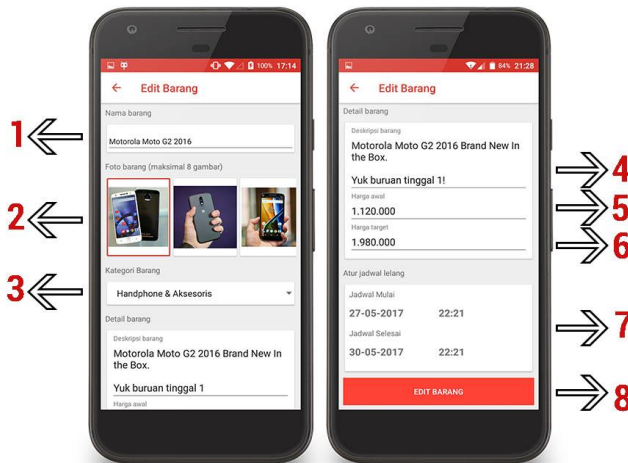


Gambar 3.25 Rancangan Antarmuka Halaman Lelang Barang

3.2.8.16 Antarmuka Halaman *Edit Barang*

Gambar 3.26 merupakan rancangan halaman edit barang. Halaman edit barang digunakan untuk melakukan perubahan informasi pada barang yang dilelang. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.26.

1. *EditText* untuk menampilkan dan memasukkan nama barang;
2. *RecyclerView* yang berisi *CardView* untuk memasukkan dan menampilkan gambar-gambar barang yang dimasukkan pengguna;
3. *Spinner* untuk menampilkan daftar kategori
4. *EditText* untuk menampilkan dan memasukkan deskripsi barang;
5. *EditText* untuk menampilkan dan memasukkan harga pembuka;
6. *EditText* untuk menampilkan dan memasukkan harga ekspektasi;
7. *Clickable TextView* untuk menampilkan dan memasukkan waktu mulai dan waktu selesai lelang; dan
8. *Button* untuk mengirim data barang ke server.



Gambar 3.26 Rancangan Antarmuka Halaman *Edit Barang*

3.2.8.17 Antarmuka Halaman Detail Barang

Halaman detail barang digunakan untuk menampilkan informasi suatu barang yang sedang dilelang. Terdapat dua versi dari halaman detail barang, yaitu halaman detail barang untuk pengguna umum dan halaman detail barang untuk pelemang.

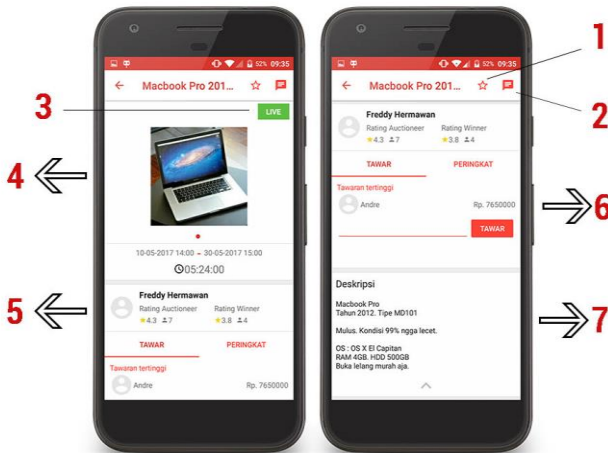
Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.27.

1. Menu untuk memasukkan ke dalam daftar barang favorit.
2. Menu untuk menghubungi pelemang melalui fitur pesan singkat atau *chat*.
3. *Fragment* yang menampilkan status lelang barang;
4. *ViewPager* yang menampilkan gambar-gambar barang.
5. *Fragment* yang menampilkan detail pelemang;
6. *Fragment* yang menampilkan aktivitas lelang;

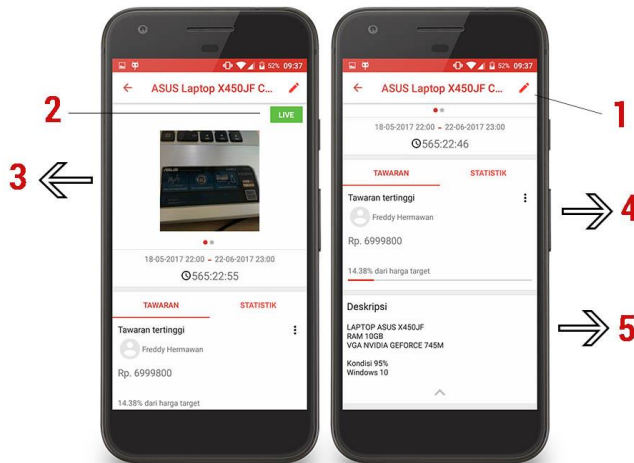
7. *Fragment* yang menampilkan deskripsi barang yang dilelang.

Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.28.

1. Menu untuk membuka halaman edit barang;
2. *Fragment* yang menampilkan status lelang barang;
3. *ViewPager* yang menampilkan gambar-gambar barang;
4. *Fragment* yang menampilkan aktivitas lelang;
5. *Fragment* yang menampilkan deskripsi barang yang dilelang.



Gambar 3.27 Rancangan Antarmuka Halaman Detail Barang untuk Pengguna Umum

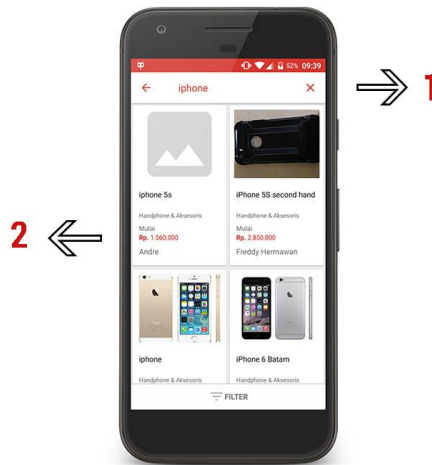


Gambar 3.28 Rancangan Antarmuka Halaman Detail Barang untuk Pelelang

3.2.8.18 Antarmuka Halaman Pencarian

Gambar 3.29 merupakan rancangan halaman pencarian. Halaman pencarian digunakan untuk mencari barang berdasarkan kata kunci. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.29.

1. *SearchView* pada toolbar untuk memasukkan kata kunci;
2. *Fragment* yang digunakan untuk menampilkan hasil pencarian saat kata kunci sudah dimasukkan.

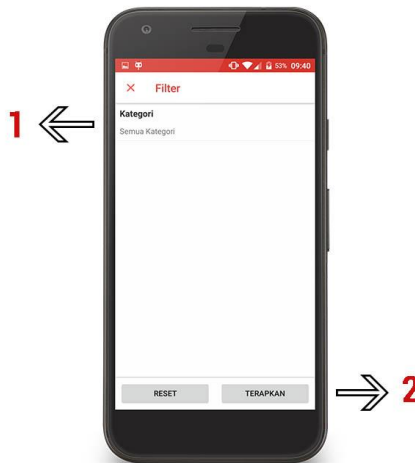


Gambar 3.29 Rancangan Antarmuka Halaman Pencarian

3.2.8.19 Antarmuka Halaman Penyaringan Pencarian

Gambar 3.30 merupakan rancangan halaman penyaringan pencarian. Halaman penyaringan pencarian digunakan untuk melakukan penyaringan terhadap hasil pencarian dengan parameter yang sudah ditetapkan oleh pengguna pada sebuah kata kunci. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.30.

1. *RecyclerView* berisi daftar parameter yang dapat disaring. Untuk saat ini, parameter yang digunakan adalah parameter kategori.
2. Tombol untuk menerapkan penyaringan hasil pencarian.

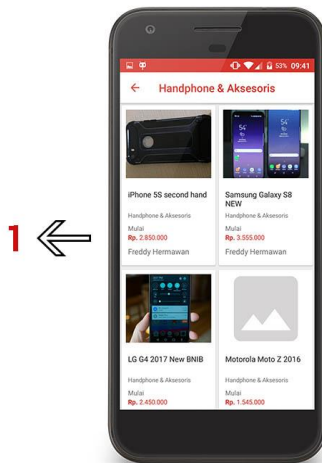


Gambar 3.30 Rancangan Antarmuka Halaman Penyaringan Pencarian

3.2.8.20 Antarmuka Halaman Kategori

Gambar 3.31 merupakan rancangan halaman kategori. Halaman kategori digunakan untuk menampilkan seluruh barang yang merupakan bagian dari kategori. Berikut beberapa penjelasan masing-masing komponen pada Gambar 3.31.

1. *RecyclerView* yang menampilkan daftar barang yang merupakan bagian dari sebuah kategori.

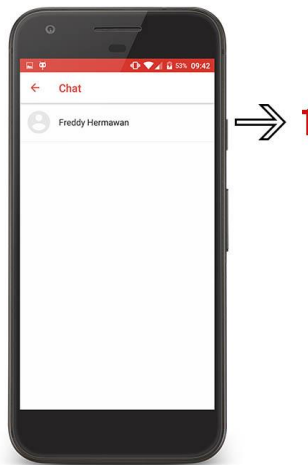


Gambar 3.31 Rancangan Antarmuka Halaman Kategori

3.2.8.21 Antarmuka Halaman Daftar *Chat*

Gambar 3.32 merupakan rancangan halaman daftar chat. Halaman daftar chat menampilkan daftar nama pengguna yang telah berinteraksi dengan pengguna. Berikut penjelasan komponen pada Gambar 3.32.

1. *RecyclerView* yang menampilkan daftar nama pengguna yang pernah berinteraksi.



Gambar 3.32 Rancangan Antarmuka Halaman Daftar Chat

3.2.8.22 Antarmuka Halaman Ruang Chat

Gambar 3.33 merupakan rancangan halaman ruang chat. Halaman ruang chat digunakan untuk mengirim pesan singkat kepada pengguna lainnya. Pengguna juga dapat melihat daftar pesan yang telah dikirim sebelumnya. Berikut penjelasan beberapa komponen yang terdapat pada Gambar 3.33.

1. *RecyclerView* yang menampilkan pesan-pesan antar kedua pengguna
2. *EditText* untuk memasukkan pesan singkat yang akan dikirimkan
3. Ikon untuk mengirim pesan

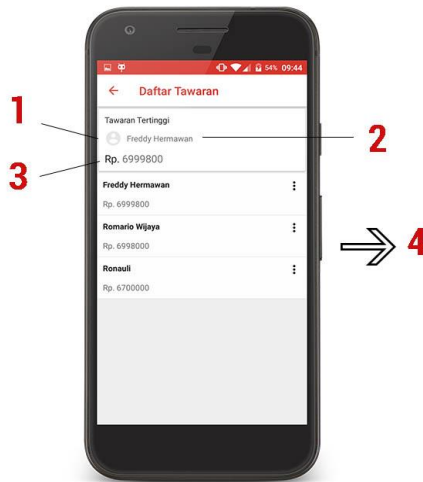


Gambar 3.33 Rancangan Antarmuka Halaman Ruang Chat

3.2.8.23 Antarmuka Halaman Daftar Tawaran

Gambar 3.34 merupakan rancangan halaman daftar tawaran. Halaman daftar tawaran digunakan untuk melihat daftar tawaran yang masuk. Berikut penjelasan beberapa komponen yang terdapat pada Gambar 3.34.

1. *ImageView* menampilkan avatar penawar tertinggi
2. *TextView* menampilkan nama penawar tertinggi
3. *TextView* menampilkan harga penawaran tertinggi
4. *RecyclerView* menampilkan daftar tawaran yang masuk dari pengguna yang telah melakukan penawaran

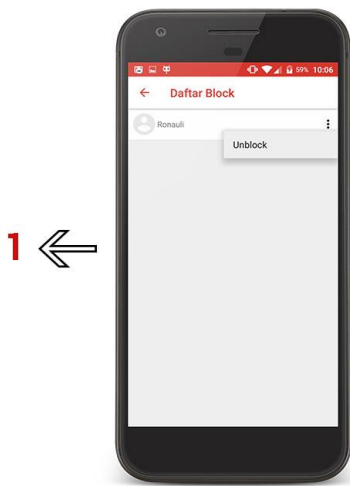


Gambar 3.34 Rancangan Antarmuka Halaman Daftar Tawaran

3.2.8.24 Antarmuka Halaman Daftar Blok

Gambar 3.35 merupakan rancangan halaman daftar blok. Halaman daftar blok digunakan untuk menampilkan pengguna yang telah di blok oleh pelelang pada suatu sesi lelang. Berikut penjelasan beberapa komponen yang terdapat pada Gambar 3.35.

1. *RecyclerView* yang menampilkan nama penawar yang telah di blok



Gambar 3.35 Rancangan Antarmuka Halaman Daftar Blok

Tabel 3.4 Daftar Atribut Tabel *Users*

Nama Kolom	Tipe Data	Keterangan
id	integer	<i>Primary key</i> dari tabel users.
name	character varying(255)	Nama pengguna.
email	character varying(255)	Email pengguna.
password	character varying(255)	Password pengguna.
address	character varying(255)	Alamat pengguna.
phone	character varying(255)	Nomor telepon pengguna.
id_city	integer	<i>Foreign key</i> dari tabel cities.
id_province	integer	<i>Foreign key</i> dari tabel provinces.
created_at	timestamp	Tanggal profil dibuat pertama kali.
updated_at	timestamp	Tanggal profil terakhir diperbaharui.
username	character varying(50)	Username pengguna.
domain	character varying(20)	Domain untuk URL milik pengguna.
remember_token	character varying(100)	Token untuk laravel. Tidak digunakan pada tugas akhir ini.
verification	timestamp	Tidak digunakan pada tugas akhir ini.
avatar_url	text	URL gambar pengguna.

Tabel 3.5 Daftar Atribut Tabel *Items* (1)

Nama Kolom	Tipa Data	Keterangan
id	big integer	<i>Primary key</i> dari tabel items.
id_user	integer	<i>Foreign key</i> dari tabel users.
name	character varying(255)	Nama item.
description	text	Deskripsi item.
starting_price	bigint	Harga pembuka lelang.
expected_price	bigint	Harga yang ditarget pelelang.
start_time	timestamp	Waktu mulai lelang.
end_time	timestamp	Waktu selesai lelang.
id_category	integer	<i>Foreign key</i> dari tabel categories.
winner_chosen_status	boolean	Status untuk mengetahui apakah pemenang lelang telah terpilih.
id_user_winner	integer	ID pengguna yang memimpin lelang / memenangkan lelang.
delete_status	boolean	Status apakah item dihapus oleh pengguna.
bid_time	integer	Jumlah sesi lelang yang telah dilakukan terhadap item yang bersangkutan.
bid_status	integer	Status lelang, apakah belum dimulai, sedang dimulai, maupun sudah selesai.
bid_price	bigint	Harga penawaran tertinggi / harga penawaran pemenang lelang.

Tabel 3.6 Daftar Atribut Tabel *Items* (2)

Nama Kolom	Tipe Data	Keterangan
end_time_after_extended	timestamp	Waktu selesai setelah durasi lelang diperpanjang (jika lelang sempat diperpanjang)
status_extend	integer	Status apakah durasi lelang diperpanjang atau tidak.
cancel_status	boolean	Status apakah lelang dibatalkan.

Tabel 3.7 Daftar Atribut Tabel *Bids*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> dari tabel bids.
id_user	integer	<i>Foreign key</i> dari tabel users.
id_item	big integer	<i>Foreign key</i> dari tabel items.
price_bid	big integer	Harga tawaran.
win_status	boolean	Status apakah tawaran ini terpilih sebagai pemenang atau tidak.
bid_timestamp	timestamp	Waktu tawaran diajukan.
bid_time	integer	Sesi lelang dari item yang ditawarkan.
is_cancelled	boolean	Status apakah tawaran ini dibatalkan oleh pelelang.

Tabel 3.8 Daftar Atribut Tabel *Bidlogs*

Nama Kolom	Tipe Data	Keterangan
id_item	big integer	<i>Foreign key</i> pada tabel bidlogs dan <i>foreign key</i> dari tabel items.
price_now	big integer	Harga tawaran tertinggi
id_user_bidder	integer	Menyimpan ID pengguna sebagai penawar tertinggi
id_bid	big integer	Menyimpan ID tawaran tertinggi.

Tabel 3.9 Daftar Atribut Tabel *BiddingBlock*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> pada tabel biddingblock.
id_item	big integer	<i>Foreign key</i> dari tabel items. Menyimpan ID item yang sedang dilelang.
id_user_auctioneer	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna pelelang.
id_user_blocked	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna penawar yang di blok.
block_timestamp	timestamp	Waktu dilakukan blok.
item_bid_time	integer	Sesi lelang.
is_blocked	boolean	Status apakah penawar masih di blok atau tidak.

Tabel 3.10 Daftar Atribut Tabel *Categories*

Nama Kolom	Tipe Data	Keterangan
id	integer	<i>Primary key</i> pada tabel <i>categories</i> .
name	character varying(255)	Nama kategori.
description	text	Deskripsi kategori.

Tabel 3.11 Daftar Atribut Tabel *Cities*

Nama Kolom	Tipe Data	Keterangan
id	integer	<i>Primary key</i> pada tabel <i>cities</i> .
name	character varying(255)	Nama kota.
id_province	integer	<i>Foreign key</i> dari tabel <i>provinces</i> .

Tabel 3.12 Daftar Atribut Tabel *Provinces*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> pada tabel <i>provinces</i> .
name	character varying(255)	Nama provinsi.

Tabel 3.13 Daftar Atribut Tabel *RatingLogs*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> pada tabel ratinglogs.
id_user_winner	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna pemenang.
id_user_auctioneer	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna pelelang.
id_item	big integer	<i>Foreign key</i> dari tabel items. Menyimpan ID item yang dilelang.
status_rating_from_winner	boolean	Status apakah pemenang lelang sudah memberikan rating kepada pelelang.
status_rating_from_auctioneer	boolean	Status apakah pelelang sudah memberikan rating kepada pemenang lelang.
bid_time	integer	Sesi lelang.
id_rating_bidder	big integer	Menyimpan ID dari rating yang sudah diberikan di tabel ratingbidders.
id_rating_auctioneer	big integer	Menyimpan ID dari rating yang sudah diberikan di tabel ratingauctioneers.

Tabel 3.14 Daftar Atribut Tabel *RatingAuctioneers*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> pada tabel <i>ratingauctioneers</i> .
id_user_auctioneer	integer	<i>Foreign key</i> dari tabel <i>users</i> . Menyimpan ID pengguna pelelang yang diberi ulasan.
id_user_rater	integer	<i>Foreign key</i> dari tabel <i>users</i> . Menyimpan ID pengguna pemenang yang memberikan ulasan.
nama_user_auctioneer	character varying(255)	Nama pengguna pelelang.
nama_user_rater	character varying(255)	Nama pengguna pemenang.
Rate	integer	Ulasan yang diperoleh (dalam angka 1-5).
id_item	big integer	<i>Foreign key</i> dari tabel <i>items</i> . Menyimpan ID item yang dilelang.
bid_time	integer	Sesi lelang.
rate_message	text	Ulasan yang diperoleh (dalam bentuk text).

Tabel 3.15 Daftar Atribut Tabel *RatingBidders*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> dari tabel ratingbidders.
id_user_bidder	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna pemenang yang diberi ulasan.
id_user_rater	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna pelelang yang memberi ulasan.
nama_user_bidder	character varying(255)	Nama pengguna pemenang.
nama_user_rater	character varying(255)	Nama pengguna pelelang.
rate	integer	Ulasan yang diperoleh (dalam angka 1-5).
id_item	big integer	<i>Foreign key</i> dari tabel items. Menyimpan ID item yang dilelang.
bid_time	integer	Sesi lelang.
rate_message	text	Ulasan yang diperoleh (dalam bentuk text).

Tabel 3.16 Daftar Atribut Tabel *Notification*

Nama Kolom	Tipe Data	Keterangan
id	text	<i>Primary key</i> pada tabel notifications. Menyimpan ID unik dari perangkat keras pengguna.
id_user	integer	<i>Foreign key</i> dari tabel users. Menyimpan ID pengguna pemilik perangkat keras.
fcm_token	text	Menyimpan token Firebase Cloud Messaging.
is_logged_in	boolean	Status apakah pengguna sedang <i>login</i> atau tidak.

Tabel 3.17 Daftar Atribut Tabel *Favorites*

Nama Kolom	Tipe Data	Keterangan
id	big integer	<i>Primary key</i> pada tabel favorites.
id_user	integer	<i>Foreign key</i> pada tabel users.
id_item_favorite	big integer	<i>Foreign key</i> pada tabel items. Menyimpan ID item yang dijadikan favorit.
remove_status	boolean	Status apakah item sudah dihapus dari daftar favorit.
time_listed	timestamp	Waktu memasukkan ke daftar favorit.

Tabel 3.18 Daftar Atribut Dokumen *ItemImages*

Nama Kolom	Tipe Data	Keterangan
_id	BSON Object	<i>Primary key</i> pada collection ItemImages.
item_id	BSON String	Menyimpan id dari barang.
url	BSON String	Menyimpan URL dari item.
is_main_image	BSON boolean	Status apakah gambar merupakan gambar utama.

Tabel 3.19 Daftar Atribut Dokumen *ChatRoom*

Nama Kolom	Tipe Data	Keterangan
_id	BSON Object	<i>Primary key</i> pada collection ChatRooms.
room	BSON String	Menyimpan ID ruangan berkirim pesan.
id_user_1	BSON String	Menyimpan ID pengguna 1 yang terlibat.
id_user_2	BSON String	Menyimpan ID pengguna 2 yang terlibat.
created_at	BSON Date Object	Menyimpan tanggal dibuatnya dokumen.
updated_at	BSON Date Object	Menyimpan tanggal diperbaharainya dokumen.

Tabel 3.20 Daftar Atribut Dokumen *UserChat*

Nama Kolom	Tipe Data	Keterangan
_id	BSON Object	<i>Primary key</i> pada collection <i>UserChat</i> .
room	BSON String	Menyimpan ID ruangan berkirim pesan.
sender	BSON String	Menyimpan ID pengguna yang mengirimkan pesan.
Msg	BSON String	Menyimpan pesan yang dikirim pengguna.
Sent	BSON Date Object	Menyimpan tanggal dikirimnya pesan.

Tabel 3.21 *Request API* Modul Informasi Pengguna

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>

Tabel 3.22 *Response Body API* Modul Informasi Pengguna Bagian 1

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.23 *Response Body* API Modul Informasi Pengguna Bagian 2

Nama Kolom	Keterangan
id_user	ID pengguna.
username	Username pengguna.
nama_user	Nama pengguna.
email_user	Email pengguna.
password_user	Password pengguna.
address_user	Alamat pengguna.
phone_user	Nomor telepon pengguna.
city_name	Nama kota asal pengguna.
province_name	Nama provinsi asal pengguna.
avatar_url	Alamat URL avatar pengguna.

Tabel 3.24 *Request Body* API Modul Ubah Profil Pengguna

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
name	Nama pengguna.
phone	Nomor telepon pengguna.
email	Alamat email pengguna.
userid	ID pengguna.

Tabel 3.25 *Response Body* API Modul Ubah Profil Pengguna

Nama Kolom	Keterangan
status	Status kesuksesan dalam pemanggilan API dan perubahan yang dilakukan.
res	Status kesuksesan pembaharuan yang dilakukan.

Tabel 3.26 *Request API Modul Ubah Alamat Pengguna*

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
address	Alamat pengguna.
userid	ID pengguna.

Tabel 3.27 *Response Body API Modul Ubah Alamat Pengguna*

Nama Kolom	Keterangan
status	Status kesuksesan dalam pemanggilan API dan perubahan yang dilakukan.
result	Status kesuksesan pembaharuan yang dilakukan.

Tabel 3.28 *Request API Modul Ubah Password Pengguna*

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
userid	ID pengguna.
oldpassword	Password lama pengguna.
newpassword	Password baru pengguna.

Tabel 3.29 *Response Body API Modul Ubah Password Pengguna*

Nama Kolom	Keterangan
status	Status kesuksesan dalam pemanggilan API dan perubahan yang dilakukan.
result	Status kesuksesan pembaharuan yang dilakukan.

Tabel 3.30 Request API Modul Informasi Barang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
itemid	ID item.

Tabel 3.31 Response Body API Modul Informasi Barang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.32 Response Body API Modul Informasi Barang

Nama Kolom	Keterangan
items_id	ID barang.
items_name	Nama barang.
item_description	Deskripsi barang.
starting_price	Harga awal/pembuka.
expected_price	Harga ekspektasi/target.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
id_category	ID kategori barang.
user_name	Nama pelemang.
url	Kumpulan alamat URL dari gambar item.

Tabel 3.33 Request API Modul Lelang Barang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
id_user	ID pengguna yang melelang barang.
nama_user	Nama pengguna yang melelang barang.
name	Nama barang.
description	Deksripsi barang.
starting_price	Harga awal/pembuka.
expected_price	Harga ekspektasi/target.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
id_category	ID kategori.
nama_category	Nama kategori.
user_domain	Domain milik pengguna.

Tabel 3.34 Response Body API Modul Lelang Barang

Nama Kolom	Keterangan
status	Status kesuksesan pemanggilan API.
id_item	ID dari barang yang telah dimasukkan.

Tabel 3.35 Request API Modul Lelang Barang

Nama Kolom	Keterangan
item_id	ID dari barang yang akan diedit.
token	Token pengguna yang didapat setelah <i>login</i>
id_user	ID pengguna yang melelang barang.
nama_user	Nama pengguna yang melelang barang.
name	Nama barang.
description	Deksripsi barang.
starting_price	Harga awal/pembuka.
expected_price	Harga ekspektasi/target.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
id_category	ID kategori.
nama_category	Nama kategori.
user_domain	Domain milik pengguna.

Tabel 3.36 Response Body API Modul Lelang Barang

Nama Kolom	Keterangan
status	Status kesuksesan pemanggilan API dan pembaharuan yang dilakukan.
res	Status kesuksesan pembaharuan yang dilakukan.

Tabel 3.37 Request API Modul Hapus Barang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
item_id	ID barang yang akan dihapus.

Tabel 3.38 *Response Body* API Modul Hapus Barang

Nama Kolom	Keterangan
status	Status kesuksesan pemanggilan API dan penghapusan yang dilakukan.

Tabel 3.39 *Request* API Modul Mendapatkan Daftar Barang Milik Pengguna

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .

Tabel 3.40 *Response Body* API Modul Mendapatkan Daftar Barang Milik Pengguna (Bagian 1)

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.41 *Response Body* API Modul Mendapatkan Daftar Barang Milik Pengguna (Bagian 1)

Nama Kolom	Keterangan
items_id	ID barang.
items_name	Nama barang.
item_description	Deskripsi barang.
starting_price	Harga awal/pembuka.
expected_price	Harga ekspektasi/target.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
user_name	Nama pengguna.
url	Alamat URL dari gambar barang.

Tabel 3.42 Request API Modul Mendapatkan Daftar Ulasan untuk Pelelang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .

Tabel 3.43 Response Body API Modul Mendapatkan Daftar Ulasan untuk Pelelang (Bagian 1)

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.44 Response Body API Modul Mendapatkan Daftar Ulasan untuk Pelelang (Bagian 2)

Nama Kolom	Keterangan
id_rating	ID dari rating yang diberikan, <i>primary key</i> dari tabel ratingauctioneers.
id_rater	ID dari pengguna yang memberikan rating.
nama_rater	Nama pengguna yang memberikan rating.
id_item	ID barang yang dilelang.
nama_item	Nama barang yang dilelang.
bid_time	Sesi lelang barang.
rate	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.45 Request API Modul Mendapatkan Daftar Ulasan sebagai Pemenang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .

Tabel 3.46 *Response Body* API Modul Mendapatkan Daftar Ulasan sebagai Pemenang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.47 *Response Body* API Modul Mendapatkan Daftar Ulasan sebagai Pemenang

Nama Kolom	Keterangan
id_rating	ID dari rating yang diberikan, <i>primary key</i> dari tabel ratingbidders.
id_rater	ID dari pengguna yang memberikan rating.
nama_rater	Nama pengguna yang memberikan rating.
id_item	ID barang yang dilelang.
nama_item	Nama barang yang dilelang.
bid_time	Sesi lelang barang.
Rate	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.48 *Request* API Modul Mendapatkan Daftar Ulasan yang Dapat Diberikan kepada Pemenang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>

Tabel 3.49 *Response Body* API Modul Mendapatkan Daftar Ulasan yang Dapat Diberikan kepada Pemenang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.50 *Response Body* API Modul Mendapatkan Daftar Ulasan yang Dapat Diberikan kepada Pemenang

Nama Kolom	Keterangan
id_ratinglogs	<i>Primary key</i> dari tabel ratinglogs.
id_item	ID barang yang dilelang.
id_user_winner	ID pemenang lelang.
nama_user_winner	Nama dari pemenang lelang.
status_rating_from_auctioneer	Status apakah rating telah diberikan oleh pelelang kepada pemenang.
nama_item	Nama dari barang yang dilelang.
bid_time_item	Sesi lelang barang.

Tabel 3.51 *Request* API Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>

Tabel 3.52 *Response Body* API Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.53 *Response Body* API Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang

Nama Kolom	Keterangan
id_ratinglogs	<i>Primary key</i> dari tabel ratinglogs.
id_item	ID barang yang dilelang.
id_user_auctioneer	ID pelelang barang.
nama_user_auctioneer	Nama dari pelelang barang.
status_rating_from_winner	Status apakah rating telah diberikan oleh pemenang kepada pelelang.
nama_item	Nama dari barang yang dilelang.
bid_time_item	Sesi lelang barang.

Tabel 3.54 *Request* API Modul Memasukkan Ulasan Baru untuk Pemenang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
id_bidder	ID dari pemenang lelang.
id_rater	ID dari pemberi ulasan (dalam hal ini pelelang barang).
id_item	ID barang yang dilelang.
bid_time	Sesi lelang barang.
rate_given	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.55 *Response Body* API Modul Informasi Pengg Memasukkan Ulasan Baru untuk Pemenang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API dan memasukkan ulasan baru.
res	Status kesuksesan eksekusi API dalam angka.

Tabel 3.56 Request API Modul Memasukkan Ulasan Baru untuk Pelelang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
id_auctioneer	ID dari pelelang barang.
id_rater	ID dari pemberi ulasan (dalam hal ini pemenang lelang barang).
id_item	ID barang yang dilelang.
bid_time	Sesi lelang barang.
rate_given	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.57 Response Body API Modul Memasukkan Ulasan Baru untuk Pelelang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API dan memasukkan ulasan baru.
res	Status kesuksesan eksekusi API dalam angka.

Tabel 3.58 Request API Modul Memperbaharui Ulasan untuk Pemenang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
id_bidder	ID dari pemenang lelang.
id_rater	ID dari pemberi ulasan (dalam hal ini pelelang barang).
id_item	ID barang yang dilelang.
bid_time	Sesi lelang barang.
rate_given	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.59 *Response Body* API Modul Memperbaharui Ulasan untuk Pemenang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API dan pembaharuan ulasan baru.
res	Status kesuksesan eksekusi API dalam angka.

Tabel 3.60 *Request* API Modul Memperbaharui Ulasan untuk Pelelang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
id_auctioneer	ID dari pelelang barang.
id_rater	ID dari pemberi ulasan (dalam hal ini pemenang lelang barang).
id_item	ID barang yang dilelang.
bid_time	Sesi lelang barang.
rate_given	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.61 *Response Body* API Modul Memperbaharui Ulasan untuk Pelelang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API dan pembaharuan ulasan baru.
Res	Status kesuksesan eksekusi API dalam angka.

Tabel 3.62 *Request* API Modul Mendapatkan Detail Ulasan untuk Pemenang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
ratinglogsId	ID yang merupakan <i>primary key</i> dari tabel ratinglogs.

Tabel 3.63 *Response Body* API Modul Mendapatkan Detail Ulasan untuk Pemenang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.64 *Response Body* API Modul Mendapatkan Detail Ulasan untuk Pemenang

Nama Kolom	Keterangan
id_ratinglogs	ID yang merupakan <i>primary key</i> dari tabel ratinglogs.
id_rating_winner	ID pada tabel ratingbidders.
id_user_winner	ID pemenang lelang.
nama_winner	Nama pemenang lelang.
id_item	ID barang yang dimenangkan.
nama_item	Nama barang yang dimenangkan.
bid_time_item	Sesi lelang barang.
rate	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.65 *Request* API Modul Mendapatkan Detail Ulasan untuk Pelelang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i>
ratinglogsId	ID yang merupakan <i>primary key</i> dari tabel ratinglogs.

Tabel 3.66 *Response Body* API Modul Mendapatkan Detail Ulasan untuk Pelelang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.67 *Response Body* API Modul Mendapatkan Detail Ulasan untuk Pelelang

Nama Kolom	Keterangan
id_ratinglogs	ID yang merupakan <i>primary key</i> dari tabel ratinglogs.
id_rating_auctioneer	ID pada tabel ratingauctioneers
id_user_auctioneer	ID pelelang barang.
nama_auctioneer	Nama pelelang barang.
id_item	ID barang yang dilelangkan.
nama_item	Nama barang yang dilelangkan.
bid_time_item	Sesi lelang barang.
Rate	Ulasan berupa angka yang diberikan.
rate_message	Ulasan berupa kata-kata yang diberikan.

Tabel 3.68 *Request* API Modul Mendapatkan Seluruh Daftar Riwayat Penawaran

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
userId	ID pelelang barang.

Tabel 3.69 *Response Body* API Modul Mendapatkan Seluruh Daftar Riwayat Penawaran

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.70 *Response Body* API Modul Mendapatkan Seluruh Daftar Riwayat Penawaran

Nama Kolom	Keterangan
id_bid	ID tawaran yang diberikan. <i>Primary key</i> dari tabel bids.
id_item	ID barang yang ditawarkan.
id_user	ID pelelang barang.
nama_user	Nama pelelang barang.
nama_item	Nama barang yang ditawarkan.
bid_status	Status lelang barang.
bid_time	Sesi lelang barang.
win_status	Status kemenangan lelang pada sebuah sesi lelang.
bid_timestamp	Waktu tawaran diberikan.
price_bid	Nominal tawaran yang diberikan.

Tabel 3.71 *Request* API Modul Mendapatkan Detail suatu Riwayat Penawaran

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
userId	ID dari penawar.
itemId	ID dari barang yang ditawarkan.
bidTime	Sesi lelang barang.

Tabel 3.72 *Response Body* API Modul Mendapatkan Detail suatu Riwayat Penawaran

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
Data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.73 *Response Body* API Modul Mendapatkan Detail suatu Riwayat Penawaran

Nama Kolom	Keterangan
id_bid	ID tawaran yang diberikan. <i>Primary key</i> dari tabel bids.
id_item	ID barang yang ditawarkan.
nama_item	Nama barang yang ditawarkan.
nama_user_bidder	Nama penawar barang.
price_bid	Nominal tawaran yang diberikan.
bid_time	Sesi lelang barang.
win_status	Status kemenangan lelang pada sebuah sesi lelang.
bid_timestamp	Waktu tawaran diberikan.

Tabel 3.74 *Request* API Modul Mendapatkan Status Favorit suatu Barang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
userId	ID pengguna.
itemId	ID barang.

Tabel 3.75 *Response Body* API Modul Mendapatkan Status Favorit suatu Barang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
Data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.76 *Response Body* API Modul Mendapatkan Status Favorit suatu Barang

Nama Kolom	Keterangan
id_favorite	ID favorit. <i>Primary key</i> dari tabel favorit.
status_favorite	Status favorit suatu barang bagi pengguna.

Tabel 3.77 *Request* API Modul Mendapatkan Daftar Barang Favorit

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
userId	ID pengguna.

Tabel 3.78 *Response Body* API Modul Mendapatkan Daftar Barang Favorit

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
Data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.79 *Response Body* API Modul Mendapatkan Daftar Barang Favorit

Nama Kolom	Keterangan
id_favorite	ID favorit. <i>Primary key</i> dari tabel favorit.
id_item_favorite	ID barang yang telah dijadikan favorit oleh pengguna.
nama_item	Nama barang yang telah dijadikan favorit oleh pengguna.
id_user_auctioneer	ID pelelang barang.
nama_user_auctioneer	Nama pelelang barang.
time_listed	Waktu suatu barang dijadikan barang favorit oleh pengguna.
imageurl	Alamat URL gambar barang.

Tabel 3.80 *Request* API Modul Memasukkan Barang sebagai Favorit

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
id_user	ID pengguna.
id_item	ID barang.

Tabel 3.81 *Response Body* API Modul Memasukkan Barang sebagai Favorit

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
res	Status kesuksesan eksekusi API dalam angka.

Tabel 3.82 Request API Modul Menghapus Barang dari Favorit

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
id_user	ID pengguna.
id_item	ID barang.
id_favorite	ID favorit. <i>Primary key</i> dari tabel favorit.

Tabel 3.83 Response Body API Modul Menghapus Barang dari Favorit

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
Res	Status kesuksesan eksekusi API dalam angka.

Tabel 3.84 Request API Modul Mendapatkan Daftar Ruang Chat

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .

Tabel 3.85 Response Body API Modul Mendapatkan Daftar Ruang Chat

Nama Kolom	Keterangan
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.86 *Response Body* API Modul Mendapatkan Daftar Ruang Chat

Nama Kolom	Keterangan
_id	ID dari dokumen MongoDB.
room	Ruangan pesan singkat yang telah dibuat.
id_user_1	ID pengguna 1 yang terlibat.
id_user_2	ID pengguna 2 yang terlibat.
createdAt	Tanggal terbentuknya ruangan pesan singkat.
updatedAt	Tanggal diperbaharuinya ruangan pesan singkat.
name	Nama lawan bicara.

Tabel 3.87 *Request* API Modul Mendapatkan Daftar Tawaran suatu Barang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
itemId	ID barang.
limit	Limit daftar tawaran yang ingin dimunculkan.

Tabel 3.88 *Response Body* API Modul Mendapatkan Daftar Tawaran suatu Barang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.89 *Response Body* API Modul Mendapatkan Daftar Tawaran suatu Barang

Nama Kolom	Keterangan
user_id	ID penawar.
user_name	Nama penawar.
bid_id	ID tawaran yang diberikan penawar.
bid_timestamp	Waktu tawaran yang diberikan.
bid_price	Harga tawaran yang diberikan.

Tabel 3.90 *Request* API Modul Memasukkan Token FCM

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
id_device	ID perangkat keras pengguna.
id_user	ID pengguna.
fcm_token	Token firebase cloud messaging pada perangkat keras milik pengguna.

Tabel 3.91 *Response Body* API Modul Memasukkan Token FCM

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.

Tabel 3.92 *Request* API Modul Mengubah Status Login ketika *Logout*

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
id_device	ID perangkat keras pengguna.

Tabel 3.93 *Response Body* API Modul Mengubah Status Login ketika *Logout*

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.

Tabel 3.94 Request API Modul Memblok Pengguna

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
id_item	ID barang yang dilelang.
id_user_auctioneer	ID pengguna pelelang.
id_user_blocked	ID pengguna yang ingin diblok.
bid_time	Sesi lelang barang.

Tabel 3.95 Response Body API Modul Memblok Pengguna

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
block	Status kesuksesan blok.

Tabel 3.96 Request API Modul Membebaskan Pengguna

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
id_item	ID barang yang dilelang.
id_user_auctioneer	ID pengguna pelelang.
id_user_blocked	ID pengguna yang ingin diblok.
bid_time	Sesi lelang barang.

Tabel 3.97 Response Body API Modul Membebaskan Pengguna

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
unblock	Status kesuksesan <i>unblock</i> .

Tabel 3.98 Request API Modul Mendapatkan Daftar Blok

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
itemId	ID barang yang dilelang.
bidTime	Sesi lelang barang.

Tabel 3.99 Response Body API Modul Mendapatkan Daftar Blok

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.100 Response Body API Modul Mendapatkan Daftar Blok

Nama Kolom	Keterangan
id_block	ID blok. <i>Primary key</i> pada tabel biddingblock.
id_user_blocked	ID pengguna yang di blok.
nama_user_blocked	Nama pengguna yang di blok.
block_timestamp	Waktu ketika pengguna di blok.

Tabel 3.101 Request API Modul Memilih Pemenang Lelang

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .
bid_id	ID tawaran yang ingin dipilih.

Tabel 3.102 Response Body API Modul Memilih Pemenang Lelang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.

Tabel 3.103 Request API Modul Melakukan Validasi Token

Nama Kolom	Keterangan
token	Token pengguna yang didapat setelah <i>login</i> .

Tabel 3.104 Response Body API Modul Melakukan Validasi Token

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API dan validitas token.
error	Status error (jika token tidak valid).

Tabel 3.105 Request API Modul Mendapatkan Daftar Barang

Nama Kolom	Keterangan
userId	ID pengguna.

Tabel 3.106 Response Body API Modul Mendapatkan Daftar Barang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.107 *Response Body* API Modul Mendapatkan Daftar Barang

Nama Kolom	Keterangan
items_id	ID barang.
items_name	Nama barang.
item_description	Deskripsi barang.
starting_price	Harga awal/pembuka.
expected_price	Harga target/ekspektasi.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
user_name	Nama pelelang barang.
url	Alamat URL gambar barang.

Tabel 3.108 *Request* API Modul Login

Nama Kolom	Keterangan
username	Username pengguna.
password	Password pengguna.

Tabel 3.109 *Response Body* API Modul Login

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
result	Status kesuksesan eksekusi API dalam angka.
token	Token pengguna yang didapat setelah <i>login</i> .
data	Data yang diperoleh dari basis data.

Tabel 3.110 Response Body API Modul Login

Nama Kolom	Keterangan
id_user	ID pengguna yang melakukan <i>login</i>
username	Username pengguna yang melakukan <i>login</i> .
password	Password pengguna yang melakukan <i>login</i> .
name_user	Nama pengguna yang melakukan <i>login</i> .
email_user	Alamat email pengguna yang melakukan <i>login</i> .
domain_user	Domain pengguna yang melakukan <i>login</i> .

Tabel 3.111 Request API Modul Register

Nama Kolom	Keterangan
name	Nama pengguna.
email	Alamat email pengguna.
password	Password pengguna.
address	Alamat pengguna.
phone	Nomor telepon pengguna.
id_city	ID kota asal pengguna.
id_province	ID provinsi asal pengguna.
<i>username</i>	Username pengguna.
domain	Domain pengguna.

Tabel 3.112 Response Body API Modul Register

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
message	Pesan kesuksesan eksekusi API.
result	Status kesuksesan eksekusi API dalam angka.

Tabel 3.113 Request API Modul Melakukan Validasi Input Username

Nama Kolom	Keterangan
username	Username yang akan diperiksa.

Tabel 3.114 *Response Body* API Modul Melakukan Validasi Input Username

Nama Kolom	Keterangan
status	Status dari <i>username</i> yang diperiksa.
res	Status dari <i>username</i> yang diperiksa dalam angka.

Tabel 3.115 *Request* API Modul Melakukan Validasi Input Domain

Nama Kolom	Keterangan
domain	Domain yang akan diperiksa.

Tabel 3.116 *Response Body* API Modul Melakukan Validasi Input Domain

Nama Kolom	Keterangan
status	Status dari domain yang diperiksa.
res	Status dari domain yang diperiksa dalam angka.

Tabel 3.117 *Request* API Modul Mendapatkan Daftar Ulasan sebagai Pemenang

Nama Kolom	Keterangan
userId	ID pengguna.

Tabel 3.118 *Response Body* API Modul Mendapatkan Daftar Ulasan sebagai Pemenang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.119 *Response Body* API Modul Mendapatkan Daftar Ulasan sebagai Pemenang

Nama Kolom	Keterangan
id_rating	ID rating. <i>Primary key</i> dari tabel ratingbidders.
id_rater	ID pengguna pemberi ulasan.
nama_rater	Nama pengguna pemberi ulasan.
id_item	ID barang.
nama_item	Nama barang.
bid_time	Sesi lelang barang.
rate	Ulasan dalam angka yang diberikan.
rate_message	Ulasan dalam kata-kata yang diberikan.

Tabel 3.120 *Request* API Modul Mendapatkan Daftar Ulasan sebagai Pelelang

Nama Kolom	Keterangan
userId	ID pengguna.

Tabel 3.121 *Response Body* API Modul Mendapatkan Daftar Ulasan sebagai Pelelang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.122 *Response Body* API Modul Mendapatkan Daftar Ulasan sebagai Pelelang

Nama Kolom	Keterangan
id_rating	ID rating. <i>Primary key</i> dari tabel ratingauctioneers.
id_rater	ID pengguna pemberi ulasan.
nama_rater	Nama pengguna pemberi ulasan.
id_item	ID barang.
nama_item	Nama barang.
bid_time	Sesi lelang barang.
rate	Ulasan dalam angka yang diberikan.
rate_message	Ulasan dalam kata-kata yang diberikan.

Tabel 3.123 *Request* API Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang

Nama Kolom	Keterangan
userId	ID pengguna.

Tabel 3.124 *Response Body* API Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
rate_average	Nilai rata-rata berupa angka yang diperoleh dari ulasan dalam bentuk angka yang pernah diberikan.
total_data	Total ulasan yang pernah diberikan.

Tabel 3.125 *Request* API Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang

Nama Kolom	Keterangan
userId	ID pengguna.

Tabel 3.126 *Response Body* API Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang

Nama Kolom	Keterangan
status	Status kesuksesan eksekusi API.
rate_average	Nilai rata-rata berupa angka yang diperoleh dari ulasan dalam bentuk angka yang pernah diberikan.
total_data	Total ulasan yang pernah diberikan.

Tabel 3.127 *Request* API Modul Mendapatkan Daftar Riwayat Penawaran

Nama Kolom	Keterangan
userId	ID pengguna.

Tabel 3.128 *Response Body* API Modul Mendapatkan Daftar Riwayat Penawaran

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.129 *Response Body* API Modul Mendapatkan Daftar Riwayat Penawaran

Nama Kolom	Keterangan
id_bid	ID tawaran.
id_item	ID barang.
id_user	ID pengguna pelelang barang.
nama_user	Nama pengguna pelelang barang.
nama_item	Nama barang yang dilelang.
bid_status	Status lelang barang.
bid_time	Sesi lelang barang.
win_status	Status kemenangan lelang pada sebuah sesi lelang.
bid_timestamp	Waktu tawaran diajukan.
price_bid	Harga tawaran yang diajukan.

Tabel 3.130 *Request* API Modul Mendapatkan Detail Barang

Nama Kolom	Keterangan
itemId	ID barang.

Tabel 3.131 *Response Body* API Modul Mendapatkan Detail Barang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.132 *Response Body* API Modul Mendapatkan Detail Barang

Nama Kolom	Keterangan
item_id	ID barang.
item_name	Nama barang.
item_description	Deskripsi barang.
starting_price	Harga awal/mulai.
expected_price	Harga target/ekspektasi.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
user_name	Nama pelelang barang.
user_id	ID pelelang barang.
user_id_winner	ID penawar tertinggi/pemenang lelang.
bidder_name	Nama penawar tertinggi/pemenang lelang.
item_bid_status	Status lelang barang.
item_bid_price	Harga tawaran tertinggi/harga tawaran pemenang.
bid_id_q	ID tawaran tertinggi.
bid_time	Sesi lelang barang.
winner_status	Status terpilihnya pemenang.
url	Alamat URL gambar barang.

Tabel 3.133 *Request* API Modul Mendapatkan Peringkat Penawaran pada Detail Barang

Nama Kolom	Keterangan
itemId	ID barang.
limit	Limit hasil yang ingin ditampilkan.

Tabel 3.134 *Response Body* API Modul Mendapatkan Peringkat Penawaran pada Detail Barang

Nama Kolom	Keterangan
status	Status hasil pemanggilan API.
data	Data yang diperoleh dari basis data berdasarkan kueri yang dikirim.

Tabel 3.135 *Response Body* API Modul Mendapatkan Peringkat Penawaran pada Detail Barang

Nama Kolom	Keterangan
user_id	ID pengguna penawar.
user_name	Nama pengguna penawar.
bid_id	ID tawaran.
bid_timestamp	Waktu tawaran diajukan.
bid_price	Nominal harga tawaran.

Tabel 3.136 *Request* API Modul Pencarian berdasarkan Kata Kunci

Nama Kolom	Keterangan
from	Menampilkan hasil dimulai dari indeks yang ditetapkan pada parameter <i>form</i> .
size	Menampilkan hasil sebanyak yang ditetapkan pada parameter <i>size</i> .
query	Kata kunci yang akan dicari.

Tabel 3.137 Request API Modul Pencarian berdasarkan Penyaringan

Nama Kolom	Keterangan
from	Menampilkan hasil dimulai dari indeks yang ditetapkan pada parameter <i>form</i> .
size	Menampilkan hasil sebanyak yang ditetapkan pada parameter <i>size</i> .
query	Kata kunci yang akan dicari.
params	Parameter filter yang ditetapkan.

Tabel 3.138 Request API Modul Menampilkan semua Barang pada suatu Kategori

Nama Kolom	Keterangan
from	Menampilkan hasil dimulai dari indeks yang ditetapkan pada parameter <i>form</i> .
size	Menampilkan hasil sebanyak yang ditetapkan pada parameter <i>size</i> .
query	Kata kunci yang akan dicari. Dalam hal ini adalah ID kategori.

Tabel 3.139 Response Body API Modul Pencarian

Nama Kolom	Keterangan
total	Jumlah hasil yang didapatkan.
result	Hasil pencarian yang didapatkan.

Tabel 3.140 Response Body API Modul Pencarian

Nama Kolom	Keterangan
_index	Nama indeks.
_type	Nama tipe pada indeks.
_id	ID dokumen.
_score	Angka relevansi dokumen.
_source	Dokumen yang didapatkan.

Tabel 3.141 Response Body API Modul Pencarian

Nama Kolom	Keterangan
id_item	ID barang.
id_user	ID pengguna pelelang barang.
nama_user	Nama pengguna pelelang barang.
title	Nama barang yang dilelang.
description	Deskripsi barang.
starting_price	Harga awal/mulai.
expected_price	Harga target/ekspektasi.
start_time	Waktu mulai lelang.
end_time	Waktu selesai lelang.
id_category	ID kategori yang menaungi barang.
nama_category	Nama kategori yang menaungi barang.
user_domain	Domain pengguna pelelang barang.
main_image_url	Alamat URL gambar utama barang.

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan implementasi perangkat lunak Lelangapa dengan perancangan sistem yang telah dijelaskan sebelumnya. Bahasa pemrograman yang digunakan adalah Java pada sisi aplikasi klien, dan Javascript serta PHP pada sisi server.

4.1 Lingkungan Pembangunan

Dalam proses pembangunan perangkat lunak ini digunakan beberapa perangkat pendukung baik perangkat keras maupun perangkat lunak. Lingkungan pembangunan dijelaskan pada subbab ini.

4.1.1 Lingkungan Pembangunan Perangkat Keras

Perangkat keras yang digunakan pada pembuatan tugas akhir ini terdiri dari tiga perangkat keras dengan spesifikasi sebagai berikut:

1. Laptop ASUS X450JF
 - a. Prosesor Intel® Core™ i7-4700HQ CPU @ 2.40Hz
 - b. Memori (RAM) 10GB
 - c. Sistem Operasi Windows 10 Pro (64-bit)
2. Cloud Server untuk API
 - a. Prosesor Intel® Xeon™ 1 core CPU
 - b. Memori (RAM) 512 MB
 - c. Sistem Operasi Ubuntu Server 16.04 LTS
3. Cloud Server untuk Indexing
 - a. Prosesor Intel® Xeon™ 1 core CPU
 - b. Memori (RAM) 512 MB
 - c. Sistem Operasi Ubuntu Server 16.04 LTS

4.1.2 Lingkungan Pembangunan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membuat tugas akhir ini adalah sebagai berikut:

1. Android Studio 2.3.3.
2. Atom IDE.
3. Postman.
4. Sybase PowerDesigner 15.1.
5. Adobe Photoshop CS6 64-bit.

4.2 Implementasi API

Pada subbab ini akan dibahas mengenai implementasi antarmuka pemrograman aplikasi (untuk selanjutnya disebut sebagai API) berdasarkan rancangan API yang sudah dibahas pada bab 3.

4.2.1 Implementasi API Privat

Pada subbab ini akan dibahas mengenai implementasi modul-modul API yang bersifat privat.

4.2.1.1 Modul User

Pada implementasi API modul user terdapat empat komponen sebagai berikut:

1. Modul Informasi Pengguna
Pseudocode implementasi modul informasi pengguna dapat dilihat pada Kode Sumber 4.1 dibawah ini.

```

1. function getUser(req, res) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       params = decoded.user_id;
5.       calldatabase(QUERY, ID_USER)
6.       .then(function(data) {
7.         res.json({
8.           status: "success",
9.           data : data
10.        });

```

```

11.   }
12.   .catch(function(err) {
13.     res.json ({
14.       status: "error"
15.     });
16.   }
17. }
18. } else {
19.   res.json ({
20.     status: "auth_error"
21.   });
22. }
23. }

```

Kode Sumber 4.1 Implementasi Modul Informasi Pengguna

2. Modul Ubah Profil Pengguna

Pseudocode implementasi modul ubah profil pengguna dapat dilihat pada Kode Sumber 4.2 dibawah ini.

```

1.  function updateProfile(req, res) {
2.    verifyToken(token, function(status, decoded) {
3.      if (token VERIFIED) {
4.        calldatabase QUERY_TO_UPDATE_PROFILE, [NAME,
          PHONE, EMAIL, ID_USER])
5.        .then(function(data) {
6.          res.json ({
7.            status: "success",
8.            result : "1"
9.          });
10.         }
11.        .catch(function(err) {
12.          res.json ({
13.            status: "error"
14.          });
15.        }
16.      }
17.    } else {
18.      res.json ({
19.        status: "auth_error"

```

```

20. });
21. }
22. }

```

Kode Sumber 4.2 Implementasi Modul Ubah Profil Pengguna

3. Modul Ubah Alamat Pengguna

Pseudocode implementasi modul ubah alamat pengguna dapat dilihat pada Kode Sumber 4.3 dibawah ini.

```

1. function updateAddress(req, res) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       callDatabase(QUERY_TO_UPDATE_ADDRESS, [ADDRESS,
5.         ID_USER])
6.       .then(function(data) {
7.         res.json ({
8.           status: "success",
9.           result : "1"
10.        });
11.      }
12.      .catch(function(err) {
13.        res.json ({
14.          status: "error"
15.        });
16.      }
17.    } else {
18.      res.json ({
19.        status: "auth_error"
20.      });
21.    }
22.  }

```

Kode Sumber 4.3 Implementasi Modul Ubah Alamat Pengguna

4. Modul Ubah Password Pengguna

Pseudocode implementasi modul ubah *password* pengguna dapat dilihat pada Kode Sumber 4.4 dibawah ini.

```

1. function updatePassword(req, res) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       calldatabase(QUERY_TO_UPDATE_PASSWORD, [ID_USER,
         OLD_PASSWD, NEW_PASSWD])
5.       .then(function(data) {
6.         if (UPDATE_PASSWORD_SUCCESS) {
7.           res.json ({
8.             status: "success",
9.             result : "1"
10.          });
11.        else {
12.          res.json ({
13.            status: "error-2"
14.          });
15.        }
16.      }
17.      .catch(function(err) {
18.        res.json ({
19.          status: "error"
20.        });
21.      }
22.    }
23.  } else {
24.    res.json ({
25.      status: "auth_error"
26.    });
27.  }
28. }

```

Kode Sumber 4.4 Implementasi Modul Ubah Password Pengguna

4.2.1.2 Modul Item

Pada implementasi API modul item terdapat lima komponen sebagai berikut:

1. Modul Informasi Barang

Pseudocode implementasi modul informasi barang dapat dilihat pada Kode Sumber 4.5 dibawah ini.

```

1. function.getItems (req, res, next) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_ITEM])
5.       .then (function(data) {
6.         async.eachSeries(data, function(el, next) {
7.           img.getItemImageUrl(el, function(res){
8.             el.url = res;
9.             next();
10.          })
11.        }, function(err) {
12.          if (err) {
13.            res.status(200).json({
14.              status : "error"
15.            });
16.          }
17.          else {
18.            res.status(200).json ({
19.              status: 'success',
20.              data: data
21.            });
22.          }
23.        });
24.      }).catch (function(err) {
25.        res.status(200).json({
26.          status : "exception-error"
27.        });
28.      });
29.    }
30.    else {
31.      res.status(200).json({
32.        status : "auth_error"
33.      });

```

```

34.     }
35.   });
36. }

```

Kode Sumber 4.5 Implementasi Modul Informasi Barang

2. Modul Lelang Barang

Pseudocode implementasi modul lelang barang dapat dilihat pada Kode Sumber 4.6 dibawah ini.

```

1.  function addItem(req, res) {
2.    verifyToken(token, function(status, decoded) {
3.      if (token VERIFIED) {
4.        lowBoundStartTime = req.body.start_time + 1 hour
5.        ;
6.        lowBoundNowTime = NOW_TIME + 1 hour;
7.        if (endTime isSameOrAfter lowBoundStartTime) {
8.          if (startTime isSameOrAfter lowBoundNowTime) {
9.            calldatabase(QUERY_TO_ADD_ITEMS, ITEM_INFO)
10.           .then(function(result) {
11.             indexingObject = buildIndexingObject(result);
12.             indexItems(indexingObject, function(err, resp
13.           ) {
14.             res.json({
15.               status: "success",
16.               id_item = result[0].item_id
17.             });
18.           });
19.         }
20.         .catch(function(err) {
21.           Res.json({
22.             Status: "failed"
23.           });
24.         });
25.       }
26.     }
27.   }
28.   else {
29.     res.json({
30.       status: "error",
31.       errcode : "wrong-time-start"
32.     });
33.   }
34. }

```

```

28.     });
29.   }
30. }
31. else {
32.   res.json({
33.     status: "error",
34.     errcode : "wrong-time-format"
35.   });
36. }
37. }
38. else {
39.   res.json({
40.     status: "auth_error"
41.   });
42. }
43. });

```

Kode Sumber 4.6 Implementasi Modul Lelang Barang

3. Modul *Edit* Lelang Barang

Pseudocode implementasi modul edit lelang barang dapat dilihat pada Kode Sumber 4.7 dibawah ini.

```

1. function addItem(req, res) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       lowBoundStartTime = req.body.start_time + 1 hour
5.       ;
6.       lowBoundNowTime = NOW_TIME + 1 hour;
7.       if (endTime isSameOrAfter lowBoundStartTime) {
8.         if (startTime isSameOrAfter lowBoundNowTime) {
9.           calldatabase(QUERY_TO_ADD_ITEMS, ITEM_INFO)
10.          .then(function(result) {
11.            if (UPDATE SUCCESS) {
12.              indexingObject = buildIndexingObject(result)
13.              ;
14.              indexItems(indexingObject, function(err, res
15.                p) {
16.                  res.json({

```



```

14.         status: "success",
15.         res: 1
16.     });
17. });
18. } else {
19.     res.json({
20.         status: "failed",
21.         res: 0
22.     });
23. }
24. })
25. .catch(function(err) {
26.     Res.json({
27.         Status: "failed"
28.     });
29. });
30. }
31. else {
32.     res.json({
33.         status: "error",
34.         errcode : "wrong-time-start"
35.     });
36. }
37. }
38. else {
39.     res.json({
40.         status: "error",
41.         errcode : "wrong-time-format"
42.     });
43. }
44. }
45. else {
46.     res.json({
47.         status: "auth_error"
48.     });
49. }
50. });

```

Kode Sumber 4.7 Implementasi Modul *Edit Lelang Barang*

4. Modul Hapus Barang

Pseudocode implementasi modul hapus barang dapat dilihat pada Kode Sumber 4.8 dibawah ini.

```

1. function deleteItems (req, res, next) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       calldatabase(QUERY_TO_DELETE_ITEMS, ID_ITEM)
5.       .then(function(result) {
6.         if (DELETE SUCCESS) {
7.           var delObj = {};
8.           delObj["id_item"] = req.body.item_id;
9.           deleteSingleDocument(delObj);
10.
11.           res.json({
12.             status: "success"
13.           });
14.         }
15.         else {
16.           res.json({
17.             status: "failed"
18.           });
19.         }
20.       })
21.       .catch(function(err) {
22.         res.json({
23.           status: "error"
24.         });
25.       });
26.     }
27.     else {
28.       res.json({
29.         status : "auth_error"
30.       });
31.     }
32.   });
33. }

```

Kode Sumber 4.8 Implementasi Modul Hapus Barang

5. Modul Mendapatkan Daftar Barang Milik Pengguna
Pseudocode implementasi modul mendapatkan daftar barang milik pengguna dapat dilihat pada Kode Sumber 4.9 dibawah ini.

```

1. function getItemOnUser (req, res, next){
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_USER])
5.       .then (function(data) {
6.         async.eachSeries(data, function(el, next){
7.           img.getItemImageUrl(el, function(res){
8.             el.url = res;
9.             next();
10.          });
11.        }, function(err){
12.          res.status(200).json({
13.            status: "success",
14.            data: data
15.          });
16.        });
17.      })
18.      .catch(function(err) {
19.        return next(err);
20.      });
21.    }
22.    else {
23.      res.status(200).json({
24.        status : "auth_error"
25.      });
26.    }
27.  });
28. }

```

Kode Sumber 4.9 Implementasi Modul Hapus Barang

4.2.1.3 Modul *Feedback*

Pada implementasi API modul feedback terdapat sepuluh komponen sebagai berikut:

1. Modul Mendapatkan Daftar Ulasan sebagai Pelelang
Pseudocode implementasi modul mendapatkan daftar ulasan sebagai pelelang dapat dilihat pada Kode Sumber 4.10 dibawah ini.

```

1. function getRatingAsAuctioneer(req, res, next)
2. {
3.   verifyToken(token, function(status, decoded) {
4.     if (token VERIFIED) {
5.       calldatabase(QUERY, ID_USER)
6.       .then(function(result) {
7.         res.status(200).json({
8.           status: "success",
9.           data: result
10.        });
11.      })
12.      .catch(function(err) {
13.        res.status(200).json({
14.          status: "error"
15.        });
16.      });
17.    }
18.    else {
19.      res.status(200).json({
20.        status : "auth_error"
21.      });
22.    }
23.  });
24. }

```

Kode Sumber 4.10 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pelelang

2. Modul Mendapatkan Daftar Ulasan sebagai Pemenang
Pseudocode implementasi modul mendapatkan daftar ulasan sebagai pemenang dapat dilihat pada Kode Sumber 4.11.

```

1. function getRatingAsWinner (req, res, next)
2. {
3.   verifyToken(token, function(status, decoded) {
4.     if (token VERIFIED) {
5.       calldatabase(QUERY, ID_USER)
6.       .then(function(result) {
7.         res.status(200).json({
8.           status: "success",
9.           data: result
10.        });
11.      })
12.      .catch(function(err) {
13.        res.status(200).json({
14.          status: "error"
15.        });
16.      });
17.    }
18.    else {
19.      res.status(200).json({
20.        status : "auth_error"
21.      });
22.    }
23.  });
24. }

```

Kode Sumber 4.11 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pemenang

3. Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pemenang
Pseudocode implementasi modul mendapatkan daftar ulasan yang dapat diberikan kepada pemenang dapat dilihat pada Kode Sumber 4.12 dibawah ini.

```

1. function getRatingForWinnerList(req, res, next)
2. {
3.   verifyToken(token, function(status, decoded) {
4.     if (token VERIFIED) {
5.       db.any(QUERY, ID_USER)
6.       .then(function(result) {
7.         res.status(200).json({

```

```

8.         status: "success",
9.         data: result
10.    });
11.  })
12.  .catch(function(err) {
13.    res.status(200).json({
14.      status: "error"
15.    });
16.  });
17.  }
18.  else {
19.    res.status(200).json({
20.      status : "auth_error"
21.    });
22.  }
23. });
24. }

```

Kode Sumber 4.12 Implementasi Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pemenang

4. Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang
Pseudocode implementasi modul mendapatkan daftar ulasan yang dapat diberikan kepada pelelang dapat dilihat pada Kode Sumber 4.13 dibawah ini.

```

1. function getRatingForAuctioneerList(req, res, next)
2. {
3.   verifyToken(token, function(status, decoded) {
4.     if (token VERIFIED) {
5.       db.any(QUERY, ID_USER)
6.       .then(function(result) {
7.         res.status(200).json({
8.           status: "success",
9.           data: result
10.        });
11.      })
12.      .catch(function(err) {

```

```

13.         res.status(200).json({
14.             status: "error"
15.         });
16.     });
17. }
18. else {
19.     res.status(200).json({
20.         status : "auth_error"
21.     });
22. }
23. });
24. }

```

Kode Sumber 4.13 Implementasi Modul Mendapatkan Daftar Ulasan yang dapat diberikan kepada Pelelang

5. Modul Memasukkan Ulasan Baru untuk Pemenang
Pseudocode implementasi modul memasukkan ulasan baru untuk pemenang dapat dilihat pada Kode Sumber 4.14 dibawah ini.

```

1. function insertRatingForWinner(req, res, next)
2. {
3.     verifyToken(token, function(status, decoded) {
4.         if (token VERIFIED) {
5.             db.any(QUERY, [ID_BIDDER, ID_RATER, ID_ITEM,
6.                 BID_TIME, RATE, RATE_MESSAGE])
7.             .then(function(result) {
8.                 if (INSERT SUCCESS){
9.                     res.status(200).json({
10.                        status: "success",
11.                        res: 1
12.                    });
13.                }
14.            else {
15.                res.status(200).json({
16.                    status: "failed",
17.                    res: 0
18.                });
19.            }
20.        })
21.    })

```

```

20.     .catch(function(err) {
21.         res.status(200).json({
22.             status: "error",
23.             res: -1
24.         });
25.     });
26. }
27. else {
28.     res.status(200).json({
29.         status : "auth_error"
30.     });
31. }
32. });
33. }

```

Kode Sumber 4.14 Implementasi Modul Memasukkan Ulasan Baru untuk Pemenang

6. Modul Memasukkan Ulasan Baru untuk Pelelang
Pseudocode implementasi modul memasukkan ulasan baru untuk pelelang dapat dilihat pada Kode Sumber 4.15 dibawah ini.

```

1. function insertRatingForAuctioneer(req, res, next)
2. {
3.     verifyToken(token, function(status, decoded) {
4.         if (token VERIFIED) {
5.             db.any(QUERY, [ID_AUCTIONEER, ID_RATER,
6. ID_ITEM, BID_TIME, RATE, RATE_MESSAGE])
7.             .then(function(result) {
8.                 if (INSERT SUCCESS){
9.                     res.status(200).json({
10.                         status: "success",
11.                         res: 1
12.                     });
13.                 }
14.                 else {
15.                     res.status(200).json({
16.                         status: "failed",
17.                         res: 0

```



```

17.         });
18.     }
19. })
20. .catch(function(err) {
21.     res.status(200).json({
22.         status: "error",
23.         res: -1
24.     });
25. });
26. }
27. else {
28.     res.status(200).json({
29.         status : "auth_error"
30.     });
31. }
32. });
33. }

```

Kode Sumber 4.15 Implementasi Modul Memasukkan Ulasan Baru untuk Pelelang

7. Modul Memperbaharui Ulasan untuk Pemenang
Pseudocode implementasi modul memperbaharui ulasan untuk pemenang dapat dilihat pada Kode Sumber 4.16 dibawah ini.

```

1. function updateRatingForWinner(req, res, next)
2. {
3.     verifyToken(token, function(status, decoded) {
4.         if (token VERIFIED) {
5.             db.any(QUERY, [ID_BIDDER, ID_RATER, ID_ITEM,
6.                 BID_TIME, RATE, RATE_MESSAGE])
7.             .then(function(result) {
8.                 if (UPDATE SUCCESS){
9.                     res.status(200).json({
10.                        status: "success",
11.                        res: 1
12.                    });
13.                }
14.            } else {
15.                res.status(200).json({

```

```

15.         status: "failed",
16.         res: 0
17.     });
18.     }
19. })
20. .catch(function(err) {
21.     res.status(200).json({
22.         status: "error",
23.         res: -1
24.     });
25. });
26. }
27. else {
28.     res.status(200).json({
29.         status : "auth_error"
30.     });
31. }
32. });
33. }

```

Kode Sumber 4.16 Implementasi Modul Memperbaharui Ulasan untuk Pemenang

8. Modul Memperbaharui Ulasan untuk Pelelang
Pseudocode implementasi modul memperbaharui ulasan untuk pelelang dapat dilihat pada Kode Sumber 4.17 dibawah ini.

```

1. function updateRatingForAuctioneer(req, res, next)
2. {
3.     verifyToken(token, function(status, decoded) {
4.         if (token VERIFIED) {
5.             db.any(QUERY, [ID_AUCTIONEER, ID_RATER,
6.                 ID_ITEM, BID_TIME, RATE, RATE_MESSAGE])
7.             .then(function(result) {
8.                 if (UPDATE SUCCESS){
9.                     res.status(200).json({
10.                        status: "success",
11.                        res: 1
12.                    });
13.                }
14.            });
15.        }
16.    });
17. }

```

```

13.         else {
14.             res.status(200).json({
15.                 status: "failed",
16.                 res: 0
17.             });
18.         }
19.     })
20.     .catch(function(err) {
21.         res.status(200).json({
22.             status: "error",
23.             res: -1
24.         });
25.     });
26. }
27. else {
28.     res.status(200).json({
29.         status : "auth_error"
30.     });
31. }
32. });
33. }

```

Kode Sumber 4.17 Implementasi Modul Memperbaharui Ulasan untuk Pelelang

9. Modul Mendapatkan Detail Ulasan untuk Pemenang
Pseudocode implementasi modul mendapatkan detail ulasan untuk pemenang dapat dilihat pada Kode Sumber 4.18 dibawah ini.

```

1. function getDetailRatingForWinner(req, res, next)
2. {
3.     verifyToken(token, function(status, decoded) {
4.         if (token VERIFIED) {
5.             db.any(QUERY, [ID_RATINGLOGS])
6.             .then(function(result) {
7.                 res.status(200).json({
8.                     status: "success",
9.                     data: result
10.                });
11.            })

```

```

12.     .catch(function(err) {
13.         res.status(200).json({
14.             status: "error"
15.         });
16.     });
17. }
18. else {
19.     res.status(200).json({
20.         status : "auth_error"
21.     });
22. }
23. });
24. }

```

Kode Sumber 4.18 Implementasi Modul Mendapatkan Detail Ulasan untuk Pemenang

10. Modul Mendapatkan Detail Ulasan untuk Pelelang

Pseudocode implementasi modul mendapatkan detail ulasan untuk pelelang dapat dilihat pada Kode Sumber 4.19 dibawah ini.

```

1. function getDetailRatingForAuctioneer (req, res, ne
   xt)
2. {
3.     verifyToken(token, function(status, decoded) {
4.         if (status==true) {
5.             db.any(QUERY, [ID_RATINGLOGS])
6.             .then(function(result) {
7.                 res.status(200).json({
8.                     status: "success",
9.                     data: result
10.                 });
11.             })
12.             .catch(function(err) {
13.                 res.status(200).json({
14.                     status: "error"
15.                 });
16.             });
17.         }
18.         else {

```

```

19.     res.status(200).json({
20.         status : "auth_error"
21.     });
22. }
23. });
24. }

```

Kode Sumber 4.19 Implementasi Modul Mendapatkan Detail Ulasan untuk Pelelang

4.2.1.4 Modul Riwayat Penawaran

Pada implementasi API modul riwayat penawaran terdapat dua komponen sebagai berikut:

1. Modul Mendapatkan Seluruh Daftar Riwayat Penawaran

Pseudocode implementasi modul mendapatkan seluruh daftar riwayat penawaran dapat dilihat pada Kode Sumber 4.20 dibawah ini

```

1.  function getBiddingHistory (req, res, next){
2.    verifyToken(token, function(status, decoded) {
3.      if (token VERIFIED) {
4.        db.any(QUERY, [ID_USER])
5.        .then(function(result) {
6.          res.status(200).json({
7.            status: 'success',
8.            data: result
9.          });
10.         });
11.       .catch(function(err) {
12.         res.status(200).json({
13.           status: 'error'
14.         });
15.       });
16.     }
17.     else {
18.       res.status(200).json({
19.         status : "auth_error"
20.       });
21.     }

```

```

22. });
23. }

```

Kode Sumber 4.20 Implementasi Modul Mendapatkan Seluruh Daftar Riwayat Penawaran

2. Modul Mendapatkan Detail suatu Riwayat Penawaran
Pseudocode implementasi modul mendapatkan detail suatu riwayat penawaran dapat dilihat pada Kode Sumber 4.21 dibawah ini

```

1. function getBiddingHistoryList(req, res, next){
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_USER, ID_ITEM, BID_TIME])
5.       .then(function(result) {
6.         res.status(200).json({
7.           status: 'success',
8.           data: result
9.         });
10.      })
11.      .catch(function(err) {
12.        res.status(200).json({
13.          status: 'error'
14.        });
15.      });
16.    }
17.    else {
18.      res.status(200).json({
19.        status : "auth_error"
20.      });
21.    }
22.  });
23. }

```

Kode Sumber 4.21 Implementasi Modul Mendapatkan Detail suatu Riwayat Penawaran

4.2.1.5 Modul Favorit

Pada implementasi API modul favorit terdapat empat komponen sebagai berikut:

1. Modul Mendapatkan Status Favorit suatu Barang
Pseudocode implementasi modul mendapatkan status favorit suatu barang dapat dilihat pada Kode Sumber 4.22 dibawah ini.

```

1. function getFavoriteOnItemIDAndUserID(req,res,next)
   {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_USER, ID_ITEM])
5.       .then(function(result) {
6.         res.status(200).json({
7.           status: "success",
8.           data: result
9.         });
10.      })
11.     .catch(function(err) {
12.       res.status(200).json({
13.         status: "failed"
14.       })
15.     });
16.   } else {
17.     res.status(200).json({
18.       status : "auth_error"
19.     });
20.   });
21. }
```

Kode Sumber 4.22 Implementasi Modul Mendapatkan Status Favorit suatu Barang

2. Modul Mendapatkan Daftar Barang Favorit
Pseudocode implementasi modul mendapatkan daftar barang favorit dapat dilihat pada Kode Sumber 4.23 dibawah ini.

```

1. function getFavorite(req,res,next){
2.   verifyJWT(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_USER])
5.       .then(function(result) {
6.         async.eachSeries(result, function(element,
7.           next) {
8.             getItemImageUrl(element.id_item_favorite,
9.             function(res) {
10.               element.imageurl = res;
11.               next();
12.             });
13.           }, function(err) {
14.             res.status(200).json({
15.               status: "success",
16.               data: result
17.             });
18.           });
19.         .catch(function(err) {
20.           res.status(200).json({
21.             status: "error"
22.           });
23.         }
24.         else {
25.           res.status(200).json({
26.             status : "auth_error"
27.           });
28.         }
29.       });
30.     }
31.

```

Kode Sumber 4.23 Implementasi Modul Daftar Barang Favorit

3. Modul Memasukkan Barang sebagai Favorit.
Pseudocode implementasi modul memasukkan barang sebagai favorit dapat dilihat pada Kode Sumber 4.24 dibawah ini.


```

1.  function insertFavorite(req, res, next){
2.    verifyToken(token, function(status, decoded) {
3.      if (token VERIFIED) {
4.        db.any(QUERY, [ID_USER, ID_ITEM])
5.        .then(function(result) {
6.          if (INSERT SUCCESS){
7.            res.status(200).json({
8.              status: "success",
9.              res: 1
10.            });
11.          }
12.          else {
13.            res.status(200).json({
14.              status: "failed",
15.              res: 0
16.            });
17.          }
18.        })
19.        .catch(function(err) {
20.          res.status(200).json({
21.            status: "error",
22.            res: -1
23.          });
24.        });
25.      }
26.      else {
27.        res.status(200).json({
28.          status : "auth_error"
29.        });
30.      }
31.    });
32. }

```

Kode Sumber 4.24 Implementasi Modul Memasukkan Barang sebagai Favorit

4. Modul Menghapus Barang dari Favorit
Pseudocode implementasi modul menghapus barang dari favorit dapat dilihat pada Kode Sumber 4.25 dibawah ini.

```

1. function removeFavorite(req, res, next){
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_USER, ID_ITEM,
5.         ID_FAVORIT])
6.         .then(function(result) {
7.           if (REMOVE SUCCESS){
8.             res.status(200).json({
9.               status: "success",
10.              res: 1
11.            });
12.          }
13.          else {
14.            res.status(200).json({
15.              status: "failed",
16.              res: 0
17.            });
18.          }
19.        })
20.        .catch(function(err) {
21.          res.status(200).json({
22.            status: "error",
23.            res: -1
24.          });
25.        })
26.      else {
27.        res.status(200).json({
28.          status : "auth_error"
29.        });
30.      }
31.    });
32.  }

```

Kode Sumber 4.25 Implementasi Modul Menghapus Barang dari Favorit

4.2.1.6 Modul Pesan Singkat (*Chat*)

Pada implementasi API modul pesan singkat (*chat*) terdapat dua komponen sebagai berikut:

1. Modul Mendapatkan Daftar Ruang *Chat*

Pseudocode implementasi modul mendapatkan daftar ruang chat dapat dilihat pada Kode Sumber 4.26 dibawah ini.

```

1. function getChatRoomList(req, res, next)
2. {
3.   verifyToken(token, function(status, decoded) {
4.     if (token VERIFIED) {
5.       var id = [
6.         {id_user_1 : decoded.u_id},
7.         {id_user_2 : decoded.u_id}
8.       ];
9.       db.collection(CHAT_COLLECTION).find({ $or : i
d}).toArray(function(err, result) {
10.        async.eachSeries(result, function(el, nex
t) {
11.          if (el.id_user_1 != decoded.u_id){
12.            getName(el.id_user_1, function(err, r
es) {
13.              el.name = res;
14.              next();
15.            });
16.          }
17.          else {
18.            getName(el.id_user_2, function(err, r
es) {
19.              el.name = res;
20.              next();
21.            });
22.          }
23.        }, function(err) {
24.          res.status(200).json({
25.            data : result
26.          });
27.        });
28.      });
29.    }
30.    else {
31.      res.status(200).json({
32.        status : "auth_error"
33.      });
34.    }

```

```

35.   });
36. }

```

Kode Sumber 4.26 Implementasi Modul Mendapatkan Daftar Ruang Chat

4.2.1.7 Modul Daftar Tawaran

Pada implementasi API modul daftar tawaran terdapat satu komponen sebagai berikut:

1. Modul Mendapatkan Daftar Tawaran suatu Barang.
Pseudocode implementasi modul mendapatkan daftar tawaran suatu barang dapat dilihat pada Kode Sumber 4.27 dibawah ini.

```

1.  function getOfferListOnItem(req, res, next) {
2.    verifyToken(token, function(status, decoded) {
3.      if (token VERIFIED) {
4.        if (req.query.limit) {
5.
6.          db.any(QUERY, [ID_ITEM, LIMIT])
7.            .then(function(result) {
8.              res.status(200).json({
9.                status: "success",
10.               data: result
11.              });
12.            })
13.            .catch(function(err) {
14.              res.status(200).json({
15.                status: "failed"
16.              });
17.            });
18.          }
19.          else {
20.            res.json({
21.              status: "failed"
22.            });
23.          }
24.        }
25.        else {
26.          res.status(200).json({
27.            status : "auth_error"

```

```

28.     });
29.   }
30. });
31. }

```

Kode Sumber 4.27 Implementasi Modul Mendapatkan Daftar Tawaran suatu Barang

4.2.1.8 Modul Notifikasi

Pada implementasi API modul notifikasi terdapat dua komponen sebagai berikut:

1. Modul Memasukkan Token FCM

Pseudocode implementasi modul memasukkan token FCM dapat dilihat pada Kode Sumber 4.28 dibawah ini.

```

1.  function submitToken(req, res, next) {
2.    verifyToken(token, function(status, decoded) {
3.      if (token VERIFIED) {
4.        db.any(QUERY, [ID_DEVICE, ID_USER,
5.          FCM_TOKEN])
6.          .then(function(result) {
7.            if (SUCCESS) {
8.              res.status(200).json({
9.                status: "success"
10.             });
11.           }
12.          else {
13.            res.status(200).json({
14.              status: "failed"
15.            });
16.          }
17.        })
18.      .catch(function(err) {
19.        res.status(200).json({
20.          status: "error"
21.        });
22.      })
23.    }
24.    else {
25.      res.status(200).json({

```

```

25.         status : "auth_error"
26.     });
27. }
28. });
29. }

```

Kode Sumber 4.28 Implementasi Modul Memasukkan Token FCM

2. Modul Mengubah Status *Login* ketika *Logout*
Pseudocode implementasi modul mengubah status *login* ketika *logout* dapat dilihat pada Kode Sumber 4.29 dibawah ini.

```

1. function changeLoginStatusWhenLogout(req, res, next
   ) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, ID_DEVICE)
5.       .then(function(result) {
6.         if (SUCCESS) {
7.           res.status(200).json({
8.             status: "success"
9.           });
10.        }
11.        else {
12.          res.status(200).json({
13.            status: "failed"
14.          });
15.        }
16.      })
17.      .catch(function(err) {
18.        res.status(200).json({
19.          status: "error"
20.        });
21.      });
22.    }
23.    else {
24.      res.status(200).json({
25.        status : "auth_error"
26.      });
27.    }

```

```

28.   });
29. }

```

Kode Sumber 4.29 Implementasi Modul Mengubah Status Login ketika Logout

4.2.1.9 Modul Blok

Pada implementasi API modul blok terdapat tiga komponen sebagai berikut:

1. Modul Memblok Pengguna

Pseudocode implementasi modul memblok pengguna dapat dilihat pada Kode Sumber 4.30 dibawah ini.

```

1. function blockUserOnBiddingByAPIRequest(req, res, n
   ext) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       db.any(QUERY, [ID_ITEM, ID_AUCTIONEER,
   ID_BLOCKED, BID_TIME])
5.       .then(function(result) {
6.         if (SUCCESS) {
7.           res.status(200).json({
8.             status : "success",
9.             block : true
10.          });
11.        }
12.       else {
13.         res.status(200).json({
14.           status : "failed",
15.           block : false
16.         });
17.       }
18.     })
19.     .catch(function(err) {
20.       res.status(200).json({
21.         status : "error"
22.       });
23.     });
24.   }
25.   else {

```

```

26.     res.status(200).json({
27.         status : "auth_error"
28.     });
29.     }
30. });
31. }

```

Kode Sumber 4.30 Implementasi Modul Memblok Pengguna

2. Modul Membebaskan Pengguna

Pseudocode implementasi modul membebaskan pengguna dapat dilihat pada Kode Sumber 4.31 dibawah ini.

```

1.  function blockUserOnBiddingByAPIRequest(req, res, n
    ext) {
2.      verifyToken(token, function(status, decoded) {
3.          if (token VERIFIED) {
4.              db.any(QUERY, [ID_ITEM, ID_AUCTIONEER,
                    ID_BLOCKED, BID_TIME])
5.                  .then(function(result) {
6.                      if (SUCCESS) {
7.                          res.status(200).json({
8.                              status : "success",
9.                              unblock : true
10.                         });
11.                     }
12.                     else {
13.                         res.status(200).json({
14.                             status : "failed",
15.                             unblock : false
16.                         });
17.                     }
18.                 })
19.                 .catch(function(err) {
20.                     res.status(200).json({
21.                         status : "error"
22.                     });
23.                 });
24.             }
25.             else {

```



```

26.     res.status(200).json({
27.         status : "auth_error"
28.     });
29.     }
30. });
31. }

```

Kode Sumber 4.31 Implementasi Modul Membebaskan Pengguna

3. Modul Mendapatkan Daftar Blok

Pseudocode implementasi modul mendapatkan daftar blok dapat dilihat pada Kode Sumber 4.32 dibawah ini.

```

1.  function getBlockList(req, res, next) {
2.      verifyToken(token, function(status, decoded) {
3.          if (token VERIFIED) {
4.              db.any(QUERY, [ID_ITEM, BID_TIME])
5.              .then(function (result) {
6.                  res.status(200).json({
7.                      status : 'success',
8.                      data : result
9.                  });
10.             })
11.             .catch(function(err) {
12.                 res.status(200).json({
13.                     status : 'failed',
14.                 });
15.             });
16.         } else {
17.             res.status(403).json({
18.                 status : "auth_error"
19.             });
20.         }
21.     });

```

Kode Sumber 4.32 Implementasi Modul Mendapatkan Daftar Blok

4.2.1.10 Modul Pilih Pemenang

Pada implementasi API modul pilih pemenang terdapat satu komponen sebagai berikut:

1. Modul Memilih Pemenang Lelang

Pseudocode implementasi modul memilih pemenang lelang dapat dilihat pada Kode Sumber 4.33 dibawah ini.

```

1. function stopAndSelectBidderWinnerByAPIRequest (req
   , res, next) {
2.   verifyToken(token, function(status, decoded) {
3.     if (token VERIFIED) {
4.       stopAndSelectBidderWinner(ID_BID, function(st
       ats, result) {
5.         if (status=="1") {
6.           res.status(200).json({
7.             status : "success"
8.           });
9.         }
10.        else if (status=="0") {
11.          res.status(200).json({
12.            status : "failed"
13.          });
14.        }
15.      });
16.    }
17.    else {
18.      res.status(200).json({
19.        status : "auth_error"
20.      });
21.    }
22.  });
23. }

```

Kode Sumber 4.33 Implementasi Modul Memilih Pemenang Lelang

4.2.1.11 Modul Validasi Token

Pada implementasi API modul validasi token terdapat satu komponen sebagai berikut:

1. Modul Melakukan Validasi Token

Pseudocode implementasi modul melakukan validasi token dapat dilihat pada Kode Sumber 4.34 dibawah ini.

```

1. function validateJWTOnAPIRequest(req, res, next) {
2.   if (token EXIST) {
3.     verifyToken(token, SECRET, algorithm, function(
4.       err, decoded) {
5.       if (err) {
6.         res.status(200).json({
7.           status: 'invalid',
8.           error: err
9.         });
10.      }
11.     else {
12.       res.status(200).json({
13.         status: 'valid'
14.       });
15.     }
16.   } else {
17.     res.status(200).json({
18.       status: 'error'
19.     });
20. }

```

Kode Sumber 4.34 Implementasi Modul Melakukan Validasi Token

4.2.2 Implementasi API Publik

Pada subbab ini akan dibahas mengenai implementasi modul-modul API yang bersifat publik.

4.2.2.1 Modul Item

Pada implementasi API modul item terdapat satu komponen sebagai berikut:

1. Modul Mendapatkan Daftar Barang
Pseudocode implementasi modul mendapatkan daftar barang dapat dilihat pada Kode Sumber 4.35 dibawah ini.

```

1. function getItemOnUserPublic (req, res, next) {
2.   db.any(QUERY, [ID_USER])
3.   .then (function(data) {
4.     async.eachSeries(data, function(e1, next){

```

```

5.     getItemImageUrl(el, function(res){
6.         el.url = res;
7.         next();
8.     });
9. }, function(err){
10.     res.status(200).json({
11.         status: "success",
12.         data: data
13.     });
14. });
15. })
16. .catch(function(err) {
17.     return next(err);
18. });
19. }

```

Kode Sumber 4.35 Implementasi Modul Mendapatkan Daftar Barang

4.2.2.2 Modul User

Pada implementasi API modul user terdapat empat komponen sebagai berikut:

1. Modul *Login*

Pseudocode implementasi modul *login* dapat dilihat pada Kode Sumber 4.36 dibawah ini.

```

1. function loginUsers (req, res, next){
2.     db.any(QUERY, [USERNAME, PASSWORD])
3.     .then(function (dat) {
4.         if (dat.length > 0) {
5.             var token = signToken(dat);
6.             res.status(200).json ({
7.                 status: 'success',
8.                 result: '1',
9.                 token: token,
10.                data: dat
11.            });
12.        }
13.        else {
14.            res.status(200).json ({
15.                status: 'failed',

```

```

16.         result: '0'
17.     });
18. }
19. })
20. .catch(function (err) {
21.     return next(err);
22. });
23. }

```

Kode Sumber 4.36 Implementasi Modul Login

2. Modul Register

Pseudocode implementasi modul register dapat dilihat pada Kode Sumber 4.37 dibawah ini.

```

1. function registerUsers (req, res, next){
2.   db.any(QUERY, REGISTRATION_INPUT)
3.   .then(function (data) {
4.     res.status(200)
5.     .json ({
6.       status: 'success',
7.       message: 'user created',
8.       result: '1'
9.     });
10.  })
11.  .catch (function (err) {
12.    res.status(200).json({
13.      status: 'error'
14.    });
15.  });
16. }

```

Kode Sumber 4.37 Implementasi Modul Register

3. Modul Melakukan Validasi Input Username

Pseudocode implementasi modul melakukan validasi input *username* dapat dilihat pada Kode Sumber 4.38 dibawah ini.

```

1. function checkUsers (req, res, next){
2.   db.result(QUERY, USERNAME)
3.   .then(function(results) {
4.     if (`${results.rowCount}`>0){
5.       res.status(200).json ({
6.         status: 'exist',
7.         res: `${results.rowCount}`
8.       });
9.     }
10.    else {
11.      res.status(200).json ({
12.        status: 'available',
13.        res: `${results.rowCount}`
14.      });
15.    }
16.  })
17.  .catch (function(err) {
18.    return next(err);
19.  });
20. }

```

Kode Sumber 4.38 Implementasi Modul Melakukan Validasi Input Username

4. Modul Melakukan Validasi Input Domain
Pseudocode implementasi modul melakukan validasi input domain dapat dilihat pada Kode Sumber 4.39 dibawah ini.

```

1. function checkDomain (req, res, next) {
2.   db.result(QUERY, DOMAIN)
3.   .then(function(results) {
4.     if (`${results.rowCount}`>0){
5.       res.status(200).json ({
6.         status: 'exist',
7.         res: `${results.rowCount}`
8.       });
9.     }
10.    else {
11.      res.status(200).json ({
12.        status: 'available',

```

```

13.         res: `${results.rowCount}`
14.     });
15. }
16. })
17. .catch (function(err) {
18.     return next(err);
19. });
20. }

```

Kode Sumber 4.39 Implementasi Modul Melakukan Validasi Input Domain

4.2.2.3 Modul *Feedback*

Pada implementasi API modul feedback terdapat empat komponen sebagai berikut:

1. Modul Mendapatkan Daftar Ulasan sebagai Pemenang *Pseudocode* implementasi modul mendapatkan daftar ulasan sebagai pemenang dapat dilihat pada Kode Sumber 4.40 dibawah ini.

```

1. function getRatingAsWinnerPublic(req, res, next) {
2.     db.any(QUERY, ID_USER)
3.     .then(function(result) {
4.         res.status(200).json({
5.             status: "success",
6.             data: result
7.         });
8.     })
9.     .catch(function(err) {
10.        res.status(200).json({
11.            status: "error"
12.        });
13.    });
14. }

```

Kode Sumber 4.40 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pemenang

2. Modul Mendapatkan Daftar Ulasan sebagai Pelelang
Pseudocode implementasi modul mendapatkan daftar ulasan sebagai pelelang dapat dilihat pada Kode Sumber 4.41 dibawah ini.

```

1. function getRatingAsAuctioneerPublic(req, res, next
   ) {
2.   db.any(QUERY, ID_USER)
3.   .then(function(result) {
4.     res.status(200).json({
5.       status: "success",
6.       data: result
7.     });
8.   })
9.   .catch(function(err) {
10.    res.status(200).json({
11.      status: "error"
12.    });
13.   });
14. }

```

Kode Sumber 4.41 Implementasi Modul Mendapatkan Daftar Ulasan sebagai Pelelang

3. Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang.
Pseudocode implementasi modul mendapatkan rata-rata ulasan sebagai pemenang dapat dilihat pada Kode Sumber 4.42 dibawah ini.

```

1. function getAverageRatingAsWinnerPublic(req, res, next) {
2.   db.any(QUERY, ID_USER)
3.   .then(function(result) {
4.     if (result.length > 0) {
5.       var totalData = parseInt(result[0].total_data
6.     );
7.       var rateAverage = null;
8.       if (totalData > 0){

```



```

8.         rateAverage = parseFloat(result[0].rate_ave
    rage).toFixed(1);
9.     }
10.    res.status(200).json({
11.        status: "success",
12.        rate_average : rateAverage,
13.        total_data : totalData
14.    });
15.    }
16.    else {
17.        res.status(200).json({
18.            status: "failed"
19.        });
20.    }
21.    })
22.    .catch(function(err) {
23.        res.status(200).json({
24.            status: "error"
25.        });
26.    });
27. }

```

Kode Sumber 4.42 Implementasi Modul Mendapatkan Rata-Rata Ulasan sebagai Pemenang

4. Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang
Pseudocode implementasi modul mendapatkan rata-rata ulasan sebagai Pelelang dapat dilihat pada Kode Sumber 4.43 dibawah ini.

```

1. function getAverageRatingAsAuctioneerPublic(req, re
    s, next) {
2.    db.any(QUERY, ID_USER)
3.    .then(function(result) {
4.        if (result.length > 0) {
5.            var totalData = parseInt(result[0].total_data
        );
6.            var rateAverage = null;
7.            if (totalData > 0){
8.                rateAverage = parseFloat(result[0].rate_ave
    rage).toFixed(1);

```

```

9.      }
10.     res.status(200).json({
11.       status: "success",
12.       rate_average : rateAverage,
13.       total_data : totalData
14.     });
15.   }
16.   else {
17.     res.status(200).json({
18.       status: "failed"
19.     });
20.   }
21. })
22. .catch(function(err) {
23.   res.status(200).json({
24.     status: "error"
25.   });
26. });
27. }

```

Kode Sumber 4.43 Implementasi Modul Mendapatkan Rata-Rata Ulasan sebagai Pelelang

4.2.2.4 Modul Riwayat

Pada implementasi API modul riwayat terdapat satu komponen sebagai berikut:

1. Modul Mendapatkan Daftar Riwayat Penawaran
Pseudocode implementasi modul mendapatkan daftar riwayat penawaran dapat dilihat pada Kode Sumber 4.44 dibawah ini.

```

1. function getBiddingHistoryPublic(req, res, next) {
2.   db.any(QUERY, ID_USER)
3.   .then(function(result) {
4.     res.status(200).json({
5.       status: 'success',
6.       data: result
7.     });
8.   })
9.   .catch(function(err) {

```

```

10.     res.status(200).json({
11.         status: 'error'
12.     });
13. });
14. }

```

Kode Sumber 4.44 Implementasi Modul Mendapatkan Daftar Riwayat Penawaran

4.2.2.5 Modul Detail Item

Pada implementasi API modul detail item terdapat dua komponen sebagai berikut:

1. Modul Mendapatkan Detail Barang
Pseudocode implementasi modul mendapatkan detail barang dapat dilihat pada Kode Sumber 4.45 dibawah ini.

```

1.  function getDetailItem (req, res, next) {
2.      db.any(QUERY, ID_ITEM)
3.      .then(function(data) {
4.          async.eachSeries(data, function(element, next)
5.              {
6.                  img.getItemImageUrl(element, function(res) {
7.                      element.url = res;
8.                      element.start_time_ms = moment(element.start_time).valueOf();
9.                      element.end_time_ms = moment(element.end_time).valueOf();
10.                 });
11.             }, function(err) {
12.                 res.status(200).json({
13.                     status: 'success',
14.                     data: data,
15.                     server_time: moment(),
16.                     server_time_in_millisecond: moment().valueOf()
17.                 });
18.             })
19.         });
20.     });
21. }

```

```

19.   .catch(function(err) {
20.     return next(err);
21.   })
22. }

```

Kode Sumber 4.45 Implementasi Modul Mendapatkan Detail Barang

2. Modul Mendapatkan Peringkat Penawaran pada Detail Barang

Pseudocode implementasi modul mendapatkan peringkat penawaran pada detail barang dapat dilihat pada Kode Sumber 4.46 dibawah ini.

```

1.  function getPeringkat(req, res, next) {
2.    db.any(QUERY, [ID_ITEM, LIMIT])
3.    .then(function(result) {
4.      res.status(200).json({
5.        status: "success",
6.        data: result
7.      });
8.    })
9.    .catch(function(err) {
10.     res.status(200).json({
11.       status: "failed"
12.     });
13.   });
14. }

```

Kode Sumber 4.46 Implementasi Modul Informasi Pengguna

4.2.2.6 Modul Pencarian

Pada implementasi API modul pencarian terdapat tiga komponen sebagai berikut:

1. Modul Pencarian berdasarkan Kata Kunci
Pseudocode implementasi modul pencarian berdasarkan kata kunci dapat dilihat pada Kode Sumber 4.47 dibawah ini.

```

1. function searchByKeywordOnTitle(req, res, next) {
2.   baldur.engine.search({
3.     body : {
4.       from : FROM,
5.       size : SIZE,
6.       query : {
7.         query_string : {
8.           query : QUERY,
9.         }
10.      }}
11. }, function(error, response) {
12.   if (error) {
13.     res.status(200).json({
14.       status : 'error'
15.     });
16.   } else {
17.     res.status(200).json({
18.       total : response.hits.total,
19.       result : response.hits.hits
20.     });
21.   });
22. }

```

Kode Sumber 4.47 Implementasi Modul Pencarian berdasarkan Kata Kunci

2. Modul Pencarian berdasarkan Penyaringan
Pseudocode implementasi modul pencarian berdasarkan penyaringan dapat dilihat pada Kode Sumber 4.48 dibawah ini.

```

1. function searchByKeywordFilter(req, res, next) {
2.   baldur.engine.search({
3.     body : {
4.       from : FROM,
5.       size : SIZE,
6.       query : {
7.         bool : {
8.           must : {
9.             query_string : {
10.              query : QUERY

```

```

11.         }
12.     },
13.     filter : PARAMS
14. }
15. }
16. }
17. }, function(error, response) {
18.     if (error) {
19.         res.status(200).json({
20.             status : 'error'
21.         });
22.     } else {
23.         res.status(200).json({
24.             total : response.hits.total,
25.             result : response.hits.hits
26.         });
27.     });
28. }

```

Kode Sumber 4.48 Implementasi Modul Pencarian berdasarkan Penyaringan

3. Modul Menampilkan semua Barang pada suatu Kategori
Pseudocode implementasi modul menampilkan semua barang pada suatu kategori dapat dilihat pada Kode Sumber 4.49 dibawah ini.

```

1. function showItemsByCategory(req, res, next) {
2.     baldur.engine.search({
3.         body : {
4.             from : FROM,
5.             size : SIZE,
6.             query : {
7.                 query_string : {
8.                     fields : ["id_category"],
9.                     query : QUERY
10.                }
11.            }
12.        }
13.    }, function(error, response) {
14.        if (error) {

```

```

15.         res.status(200).json({
16.             status : 'error'
17.         });
18.     }
19.     else {
20.         res.status(200).json({
21.             total : response.hits.total,
22.             result : response.hits.hits
23.         });
24.     }
25. });
26. }

```

Kode Sumber 4.49 Implementasi Modul Menampilkan semua Barang pada suatu Kategori

4.3 Implementasi Fitur Aplikasi

Pada subbab ini akan dibahas mengenai implementasi fitur-fitur utama pada aplikasi Lelangapa berdasarkan rancangan yang sudah dibahas pada bab 3.

4.3.1 Implementasi Fitur Halaman Utama

Halaman utama menampilkan antarmuka berupa daftar kategori yang terdapat pada perangkat lunak Lelangapa. Komponen yang digunakan untuk menampilkan daftar kategori adalah sebuah *RecyclerView*. Selain itu, terdapat juga sebuah *TabLayout* yang memisahkan antara fragment halaman utama dengan fragment promo, yang mana fragment promo tidak diimplementasikan pada tugas akhir ini. Pada *toolbar* juga terdapat sebuah ikon 3 garis (*hamburger*) yang menampilkan panel navigasi dan juga ikon kaca pembesar untuk menuju ke pencarian. Untuk implementasi menampilkan muatan yang terdapat pada halaman utama dapat dilihat pada Kode Sumber 4.50. Sementara itu untuk implementasi antarmuka pada Kode Sumber 4.51. Gambar antarmuka halaman utama dapat dilihat pada Gambar 4.1.

```

1.  protected void onCreate(Bundle savedInstanceState)
    {
2.      super.onCreate(savedInstanceState);
3.      sessionManager = new SessionManager(MainActivity.this);
4.      if (sessionManager.isLoggedIn()){
5.          setContentView(R.layout.activity_main_loggedin);
6.          drawer = (DrawerLayout) findViewById(R.id.activity_main_loggedin);
7.          navigationView = (NavigationView) findViewById(R.id.nav_view_loggedin);
8.          View header = navigationView.getHeaderView(0);
9.          TextView userInfoNameNavbar = (TextView) header.findViewById(R.id.user_info_name_navbar);
10.         TextView userInfoEmailNavbar = (TextView) header.findViewById(R.id.user_info_email_navbar);
11.         userInfo = sessionManager.getSession();
12.         userInfoNameNavbar.setText(headerPrefix + userInfo.get(sessionManager.getKey_NAME()));
13.         userInfoEmailNavbar.setText(userInfo.get(sessionManager.getKey_EMAIL()));
14.     }
15.     else {
16.         setContentView(R.layout.activity_main);
17.         drawer = (DrawerLayout) findViewById(R.id.activity_main);
18.         navigationView = (NavigationView) findViewById(R.id.nav_view);
19.     }
20.     sendTokenToServer();
21.
22.     Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_main);
23.     setSupportActionBar(toolbar);
24.
25.     toggle = new ActionBarDrawerToggle(

```



```

26.         this, drawer, toolbar, R.string.nav
           igation_drawer_open, R.string.navigation_drawer_clo
           se);
27.         drawer.addDrawerListener(toggle);
28.         viewPager = (ViewPager) findViewById(R.id.v
           iewpager_main);
29.         tabLayout = (TabLayout) findViewById(R.id.t
           ab_main);
30.         setUpViewPager(viewPager);
31.         tabLayout.setupWithViewPager(viewPager);
32.
33.         setupNavigationDrawer();
34. }

```

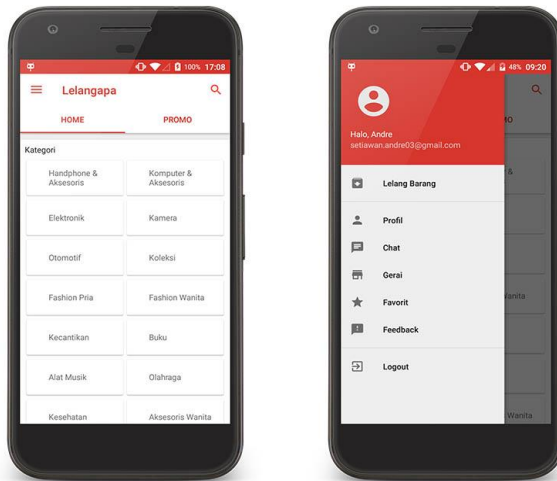
Kode Sumber 4.50 Menampilkan Muatan Halaman Utama

```

1. <android.support.v4.widget.DrawerLayout xmlns:andro
   id="http://schemas.android.com/apk/res/android"
2.     <include
3.         layout="@layout/fragment_main"
4.         android:layout_width="match_parent"
5.         android:layout_height="match_parent"
6.     />
7.     <android.support.design.widget.NavigationView
8.         android:id="@+id/nav_view"
9.         android:layout_width="wrap_content"
10.        android:layout_height="match_parent"
11.        android:layout_gravity="start"
12.        android:fitsSystemWindows="true" />
13. </android.support.v4.widget.DrawerLayout>

```

Kode Sumber 4.51 Menampilkan Halaman Utama



Gambar 4.1 Halaman Utama

4.3.2 Implementasi Fitur *Login*

Halaman *login* digunakan untuk melakukan proses *login* ke sistem. Halaman *login* terdiri dari dua buah *EditText* untuk memungkinkan pengguna memasukkan *username* dan *password*. Terdapat juga sebuah *Button* untuk memicu perangkat lunak mengakses server melalui API *login* yang telah dibuat. Jika *login* berhasil, maka server akan mengembalikan sebuah dokumen JSON berisi token yang akan disimpan di dalam memori internal *smartphone* pengguna. Token ini yang akan digunakan untuk mengakses API yang bersifat privat sebagai validasi. Untuk implementasi muatan halaman *login* dapat dilihat pada Kode Sumber 4.52 dan implementasi antarmuka halaman *login* dapat dilihat pada Kode Sumber 4.53. Gambar antarmuka halaman *login* dapat dilihat pada Gambar 4.2.

```

1. public View onCreateView(LayoutInflater inflater, V
   iewGroup container, Bundle savedInstanceState){
2.     View view = inflater.inflate(R.layout.fragment
   _login_layout, container, false);
3.     buttonLogin = (Button) view.findViewById(R.
   id.fragment_login_button);
4.     txtRegister = (TextView) view.findViewById(
   R.id.fragment_login_register_textview);
5.     editText_username = (EditText) view.findVie
   wById(R.id.fragment_login_username);
6.     editText_password = (EditText) view.findVie
   wById(R.id.fragment_login_password);
7.     textInputLayout_username = (TextInputLayout
   ) view.findViewById(R.id.input_login_layout_usernam
   e);
8.     textInputLayout_password = (TextInputLayout
   ) view.findViewById(R.id.input_login_layout_passwor
   d);
9.     setInputValidation();
10.    setHasOptionsMenu(true);
11.    return view;
12. }

```

Kode Sumber 4.52 Menampilkan Muatan Halaman *Login*

```

1. <LinearLayout
2.     android:orientation="vertical">
3.     <android.support.design.widget.TextInputLayout
4.         android:id="@+id/input_login_layout_username"
5.         android:layout_marginBottom="12dp">
6.         <EditText
7.             android:id="@+id/fragment_login_username" />
8.     </android.support.design.widget.TextInputLayout>
9.     <android.support.design.widget.TextInputLayout
10.        android:id="@+id/input_login_layout_password" >
11.        <EditText
12.            android:id="@+id/fragment_login_password" />
13.    </android.support.design.widget.TextInputLayout>
14.    <Button
15.        android:id="@+id/fragment_login_button"/>
16. <RelativeLayout>
17. <TextView

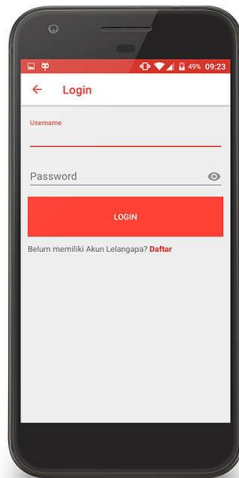
```

```

18. android:id="@+id/fragment_login_register_keterangan
   "
19. android:text="Belum memiliki Akun Lelangapa? "/>
20. <TextView
21. android:id="@+id/fragment_login_register_textview"
   />
22. </RelativeLayout>
23. </LinearLayout>

```

Kode Sumber 4.53 Menampilkan Halaman *Login*



Gambar 4.2 Halaman *Login*

4.3.3 Implementasi Fitur *Register*

Halaman antarmuka *register* digunakan untuk melakukan proses pendaftaran pengguna baru. Seluruh data yang diisikan oleh pengguna akan divalidasi terlebih dahulu di sisi *client* sebelum dikirim ke server. Untuk implementasi muatan halaman *register* dapat dilihat pada Kode Sumber 4.54 dan implementasi antarmuka halaman *register* dapat dilihat pada Kode Sumber 4.55. Gambar antarmuka halaman *register* dapat dilihat pada Gambar 4.3.

```

1. @Override
2.     public View onCreateView(LayoutInflater inflater
   , ViewGroup container, Bundle savedInstanceState){
3.         View view = inflater.inflate(R.layout.fragment_register_layout, container, false);
4.         initializeViews(view); //memuat element pada view
5.         initializeResponseListener(); //menginisiasi response listener untuk menangani respon API.
6.         initializeDataReceivers(); //menginisiasi data receiver untuk menangani respon API.
7.         geoResources = new GeoResources();
8.         setSpinnerProvince(); //memuat daftar provinsi untuk ditampilkan di spinner
9.         setupAllInputValidation(); //memberikan properti validasi input pada elemen EditText
10.        setupViewClickListener(); //memberikan properti klik pada elemen view tertentu.
11.        setHasOptionsMenu(true);
12.        return view;
13.    }

```

Kode Sumber 4.54 Menampilkan Muatan Halaman *Register*

```

1. <ScrollView
2. <LinearLayout
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical">
6.     <android.support.design.widget.TextInputLayout

```

```

7.  android:id="@+id/input_register_layout_name">
8.  <EditText
9.  android:id="@+id/fragment_register_name"
10. android:hint="Nama Lengkap" />
11. </android.support.design.widget.TextInputLayout>
12. <android.support.design.widget.TextInputLayout
13. android:id="@+id/input_register_layout_username">
14. <EditText
15. android:id="@+id/fragment_register_username"
16. android:hint="Username" />
17. </android.support.design.widget.TextInputLayout>
18. <android.support.design.widget.TextInputLayout
19. android:id="@+id/input_register_layout_password">
20. <EditText
21. android:id="@+id/fragment_register_password" />
22. </android.support.design.widget.TextInputLayout>
23. <android.support.design.widget.TextInputLayout
24. android:id="@+id/input_register_layout_domain">
25. <EditText
26. android:id="@+id/fragment_register_domain"
27. android:hint="Domain" />
28. <TextView
29. android:id="@+id/fragment_register_domain_textview"
30. android:text="https://www.lelangapa.com/" />
31. </android.support.design.widget.TextInputLayout>
32. <android.support.design.widget.TextInputLayout
33. android:id="@+id/input_register_layout_email">
34. <EditText
35. android:id="@+id/fragment_register_email" />
36. </android.support.design.widget.TextInputLayout>
37. <android.support.design.widget.TextInputLayout
38. android:id="@+id/input_register_layout_address">
39. <EditText
40. android:id="@+id/fragment_register_address" />
41. </android.support.design.widget.TextInputLayout>
42. <android.support.design.widget.TextInputLayout
43. android:id="@+id/input_register_layout_telepon">
44. <EditText
45. android:id="@+id/fragment_register_telepon" />
46. </android.support.design.widget.TextInputLayout>
47. <LinearLayout
48. android:layout_width="match_parent"
49. android:layout_height="wrap_content"

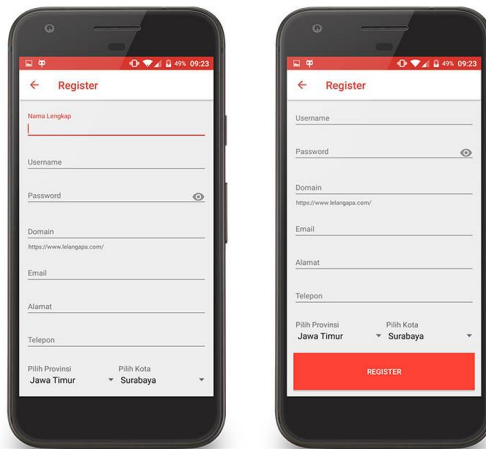
```

```

50. android:orientation="horizontal">
51. <LinearLayout
52. android:layout_width="0px"
53. android:layout_height="wrap_content"
54. android:layout_weight="1">
55. <TextView
56. android:text="Pilih Provinsi"
57. android:id="@+id/textview_provinsi"/>
58. <Spinner
59. android:id="@+id/spinner_province"
60. android:spinnerMode="dialog" />
61. </LinearLayout>
62. <LinearLayout
63. android:layout_width="0px"
64. android:layout_height="wrap_content"
65. android:orientation="vertical"
66. android:layout_weight="1">
67. <TextView
68. android:text="Pilih Kota"/>
69. <Spinner
70. android:id="@+id/spinner_city"
71. android:spinnerMode="dialog" />
72. </LinearLayout>
73. </LinearLayout>
74. <Button
75. android:id="@+id/fragment_register_button"/>
76. </LinearLayout> </ScrollView>

```

Kode Sumber 4.55 Menampilkan Halaman *Register*



Gambar 4.3 Halaman Register

4.3.4 Impelementasi Fitur Halaman Profil

Halaman profil terdiri dari sebuah *LinearLayout* yang berisi sebuah *ImageView* dan *TextView* untuk menampilkan detail pengguna yang telah melakukan *login*, dan *RecyclerView* untuk menampilkan daftar menu yang dapat diakses. Untuk implementasi muatan halaman profil dapat dilihat pada Kode Sumber 4.56 dan implementasi antarmuka halaman profil dapat dilihat pada Kode Sumber 4.57. Gambar antarmuka halaman profil dapat dilihat pada Gambar 4.4.

```

1. @Override
2.     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState){
3.         View view = inflater.inflate(R.layout.fragment_userprofile_layout, container ,false);
4.         initializeViews(view); //memuat seluruh elemen view.

```



```

5.         SessionManager sessionManager = new Session
Manager(getActivity());
6.         if (sessionManager.isLoggedIn()){
7.             setProfileHeaderText(); //menampilkan n
ama dan email pengguna yang login. }
8.             initializeAdapters(); //menginisiasi adapte
r untuk recyclerview
9.             setupRecyclerViewProperties(); //memberikan
properti untuk recyclerview yang menampilkan menu
pada fragment profil. }

```

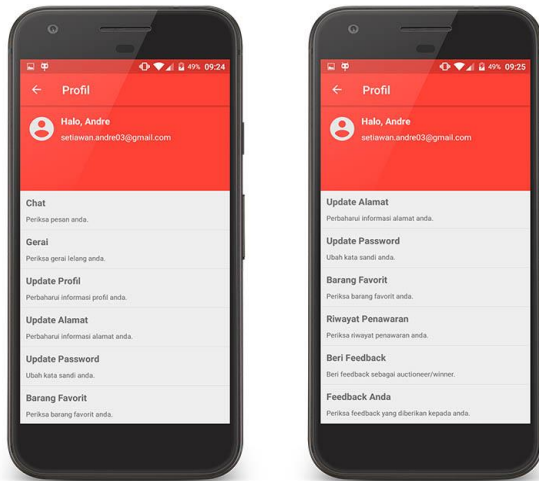
Kode Sumber 4.56 Menampilkan Muatan Halaman Profil

```

1. <LinearLayout
2.     android:id="@+id/fragment_userprofile_layout">
3.     <LinearLayout android:orientation="vertical">
4.     <LinearLayout android:orientation="horizontal">
5.     <ImageView
6.     android:id="@+id/fragment_userprofile_image_profile
"/>
7.     <LinearLayout
8.     android:orientation="vertical">
9.     <LinearLayout
10.    android:orientation="horizontal">
11.    <TextView
12.    android:text="Halo, "/>
13.    <TextView
14.    android:id="@+id/fragment_userprofile_user_name"
15.    android:text="Nama Pelelang"/>
16.    </LinearLayout>
17.    <TextView
18.    android:id="@+id/fragment_userprofile_user_email"
19.    android:text="Email pelelang"/>
20.    </LinearLayout>
21.    </LinearLayout>
22.    </LinearLayout>
23.    <android.support.v7.widget.RecyclerView
24.    android:id="@+id/userprofile_recyclerview"
25.    android:scrollbars="vertical" />
26.    </LinearLayout>

```

Kode Sumber 4.57 Menampilkan Halaman Profil



Gambar 4.4 Halaman Profil

4.3.5 Impelementasi Fitur *Edit* Profil

Halaman edit profil digunakan untuk memperbaharui data pengguna yang telah *login*. Untuk implementasi muatan halaman edit profil dapat dilihat pada Kode Sumber 4.58 dan implementasi antarmuka halaman edit profil dapat dilihat pada Kode Sumber 4.59. Gambar antarmuka halaman edit profil dapat dilihat pada Gambar 4.5.

```

1. @Override
2.     public View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
3.     {
4.         View view = inflater.inflate(R.layout.fragment_userprofile_editprofile_layout, container, false);
5.         initializeConstant(); //menginisiasi objek-objek yang akan digunakan.

```

```

5.         initializeViews(view); //memuat elemen view
           yang digunakan.
6.         initializeDataReceiver(); //menginisiasi da
           ta receiver untuk menangani respon API.
7.         initializeViewVisibilities(); //menginisias
           i visibilitas dari elemen view.
8.         setViewOnClickListener(); //memberikan
           properti klik pada elemen view tertentu.
9.         initializeAlertDialogBuilder(); //menginisi
           asi alert dialog yang akan digunakan.
10.        sessionManager = new SessionManager(getActi
           vity()); //mengambil data session pengguna.
11.        session = sessionManager.getSession();
12.        return view;
13.    }

```

Kode Sumber 4.58 Menampilkan Muatan Halaman *Edit Profil*

```

1. <android.support.v4.widget.NestedScrollView>
2. <LinearLayout android:orientation="vertical">
3. <android.support.v7.widget.CardView
4. android:layout_marginBottom="8dp">
5. <LinearLayout
6. android:orientation="vertical">
7. <TextView
8. android:text="Avatar"/>
9. <ImageView
10. android:id="@+id/fragment_usereditprofile_avatar"/>
11. <ProgressBar
12. android:id="@+id/fragment_usereditprofile_avatar_pr
   ogress_bar"/>
13. </LinearLayout>
14. </android.support.v7.widget.CardView>
15. <android.support.v7.widget.CardView
16. android:id="@+id/fragment_usereditprofile_layout_ca
   rd">
17. <ProgressBar
18. android:id="@+id/fragment_userprofile_editprofile_p
   rogress_bar_informasiakun"/>
19. <RelativeLayout
20. android:orientation="vertical">
21. <TextView

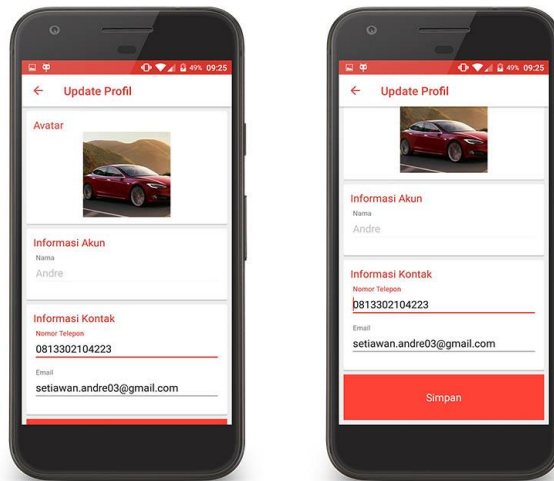
```

```

22. android:id="@+id/textview_informasi_akun"
23. android:text = "Informasi Akun" />
24. <android.support.design.widget.TextInputLayout
25. android:id="@+id/fragment_usereditprofile_name_layo
    ut">
26. <EditText
27. android:id="@+id/fragment_usereditprofile_name"
28. android:hint="Nama"/>
29. </android.support.design.widget.TextInputLayout>
30. </RelativeLayout>
31. </android.support.v7.widget.CardView>
32. <android.support.v7.widget.CardView>
33. <ProgressBar
34. android:id="@+id/fragment_userprofile_editprofile_p
    rogress_bar_informasikontak" />
35. <LinearLayout
36. android:orientation="vertical">
37. <TextView
38. android:text="Informasi Kontak"/>
39. <android.support.design.widget.TextInputLayout
40. android:id="@+id/fragment_usereditprofile_phone_lay
    out">
41. <EditText
42. android:id="@+id/fragment_usereditprofile_phone"
43. android:hint="Nomor Telepon"/>
44. </android.support.design.widget.TextInputLayout>
45. <android.support.design.widget.TextInputLayout
46. android:id="@+id/fragment_usereditprofile_email_lay
    out">
47. <EditText
48. android:id="@+id/fragment_usereditprofile_email"
49. android:hint="Email"/>
50. </android.support.design.widget.TextInputLayout>
51. </LinearLayout>
52. </android.support.v7.widget.CardView>
53. <Button
54. android:id = "@+id/fragment_usereditprofile_simpan_b
    utton"/></LinearLayout>
55. </android.support.v4.widget.NestedScrollView>

```

Kode Sumber 4.59 Menampilkan Halaman *Edit* Profil



Gambar 4.5 Halaman *Edit Profil*

4.3.6 Impelementasi Fitur *Edit Alamat*

Halaman edit alamat digunakan untuk melakukan perubahan detail alamat dari pengguna yang telah melakukan *login* sebelumnya. Untuk implementasi muatan halaman edit alamat dapat dilihat pada Kode Sumber 4.60 dan implementasi antarmuka halaman edit alamat dapat dilihat pada Kode Sumber 4.61.

```

1. @Override
2.     public View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
3.     {
4.         View view = inflater.inflate(R.layout.fragment_userprofile_editalamat_layout, container, false);
5.         //mengisi layout fragment dengan view

```

```

6.         initializeViews(view); //memuat elemen yang
           terdapat pada view.
7.         initializeDataReceivers(); //menginisiasi d
           ata receiver untuk menangani respon API.
8.         setViewOnClickListeners(); //memberikan pro
           perta klik pada elemen view tertentu.
9.         sessionManager = new SessionManager(getActi
           vity());
10.        session = sessionManager.getSession(); //me
           ngambil data session pengguna.
11.        return view;
12.    }

```

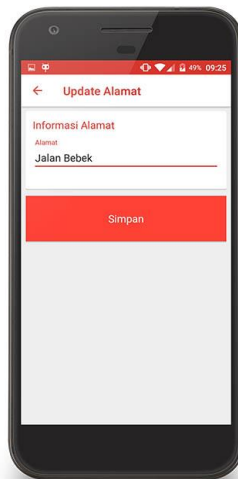
Kode Sumber 4.60 Menampilkan Muatan Halaman *Edit* Alamat

```

1. <LinearLayout android:orientation="vertical">
2.     <android.support.v7.widget.CardView>
3.         <LinearLayout
4.             android:orientation="vertical">
5.                 <TextView
6.                     android:text="Informasi Alamat"/>
7.                 <android.support.design.widget.TextInput
           tLayout>
8.                     <EditText
9.                         android:id="@+id/fragment_userp
           rofile_editalamat_alamat"
10.                        android:hint="Alamat"/>
11.                 </android.support.design.widget.TextInp
           utLayout>
12.             </LinearLayout>
13.         </android.support.v7.widget.CardView>
14.         <Button
15.             android:id="@+id/fragment_userprofile_edita
           lamat_simpan"
16.             android:text="Simpan"/>
17.     </LinearLayout>

```

Kode Sumber 4.61 Menampilkan Halaman *Edit* Alamat



Gambar 4.6 Halaman *Edit* Alamat

4.3.7 Implementasi Fitur Ubah *Password*

Halaman ubah *password* digunakan untuk melakukan perubahan *password* pengguna yang telah melakukan *login* sebelumnya. Setiap masukan *password* akan divalidasi terlebih dahulu pada sisi *client*, apakah masukan pada *password* baru dan konfirmasi *password* baru sama. Jika sama, maka data akan dikirim ke *server* untuk dilakukan validasi kedua sebelum data dikirim ke *database*. Validasi kedua adalah memeriksa apakah *password* lama yang dimasukkan adalah benar, sesuai, dan merupakan milik dari pengguna yang bersangkutan. Jika benar, maka *password* akan dirubah dan halaman akan otomatis ditutup. Jika salah, maka akan muncul pesan *error* bahwa *password* yang dimasukkan salah. Untuk implementasi muatan halaman ubah *password* dapat dilihat pada Kode Sumber 4.62 dan implementasi antarmuka halaman ubah *password* dapat dilihat pada Kode Sumber 4.63.

```

1. @Override
2.     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState){
3.
4.         View view = inflater.inflate(R.layout.fragment_userprofile_editpassword_layout, container, false);
5.         initializeViews(view); //inisiasi view.
6.         initializeDataReceiversToSubmitNewPassword(); //inisiasi data receiver untuk menerima respon API ketika password diubah.
7.         setViewClickListener(); //memberikan properti klik pada elemen view tertentu.
8.         setInputValidation(); //memberikan properti validasi input pada EditText
9.         sessionManager = new SessionManager(getActivity());
10.        session = sessionManager.getSession(); //memuat data session pengguna.
11.        return view; }

```

Kode Sumber 4.62 Menampilkan Muatan Halaman Ubah Password

```

1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     android:orientation="vertical">
3.     <android.support.v7.widget.CardView>
4.         <LinearLayout
5.             android:orientation="vertical">
6.             <TextView
7.                 android:text="Ubah Password"/>
8.             <android.support.design.widget.TextInputLayout
9.                 android:id="@+id/fragment_userprofile_editpassword_textinputlayoutoldpassword">
10.                <EditText
11.                    android:id="@+id/fragment_userprofile_editpassword_oldpassword"
12.                    android:hint="Password Lama" />
13.            </android.support.design.widget.TextInputLayout>

```

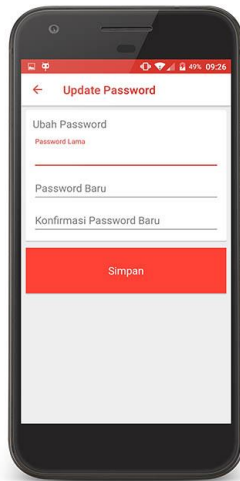


```

14.         <android.support.design.widget.TextInput
            tLayout
15.             android:id="@+id/fragment_userprofi
            le_editpassword_textinputlayoutnewpassword">
16.             <EditText
17.                 android:id="@+id/fragment_userp
            rofile_editpassword_newpassword"
18.                 android:hint="Password Baru" />
19.         </android.support.design.widget.TextInp
            utLayout>
20.         <android.support.design.widget.TextInp
            tLayout
21.             android:id="@+id/fragment_userprofi
            le_editpassword_textinputlayoutconfirmpassword">
22.             <EditText
23.                 android:id="@+id/fragment_userp
            rofile_editpassword_confirmnewpassword"
24.                 android:hint="Konfirmasi Passwo
            rd Baru" />
25.         </android.support.design.widget.TextInp
            utLayout>
26.     </LinearLayout>
27. </android.support.v7.widget.CardView>
28. <Button
29.     android:id = "@+id/fragment_userprofile_edi
            tpassword_simpan"
30.     android:text="Simpan"/>
31. </LinearLayout>

```

Kode Sumber 4.63 Menampilkan Halaman Ubah *Password*



Gambar 4.7 Halaman Ubah *Password*

4.3.8 Implementasi Fitur Beri *Feedback*

Halaman beri *feedback* digunakan untuk menampilkan daftar ulasan yang diberikan kepada pengguna lain beserta statusnya apakah sudah diberikan atau belum diberikan. Untuk implementasi halaman beri *feedback* dapat dilihat pada Kode Sumber 4.64 dan implelementasi antarmuka halaman beri *feedback* dapat dilihat pada Kode Sumber 4.65.

```

1. @Override
2.     protected void onCreate(Bundle savedInstanceState)
3.     {
4.         super.onCreate(savedInstanceState);
5.         initializeFragments(); //inisiasi fragment a
6.         initializeViews(); //inisiasi view.
7.     }

```

```

8.     private void initializeViews()
9.     {
10.        ... //memuat elemen-elemen view
11.        setupViewPager(); //inisiasi viewpager
12.        tabLayout_beriFeedback.setupWithViewPager(vi
ewPager_beriFeedback); //memasang viewpager dengan
        tablayout
13.    }
14.    private void initializeFragments()
15.    {
16.        winnerFragment = new WinnerFragment();
17.        auctioneerFragment = new AuctioneerFragment(
    );
18.    }
19.    private void setupViewPager()
20.    {
21.        FeedbackViewPagerAdapter pagerAdapter = new
FeedbackViewPagerAdapter(getSupportFragmentManager(
    ));
22.        pagerAdapter.addFragment(winnerFragment, "Pe
menang");
23.        pagerAdapter.addFragment(auctioneerFragment,
        "Pelelang");
24.        viewPager_beriFeedback.setAdapter(pagerAdapt
er);
25.    }

```

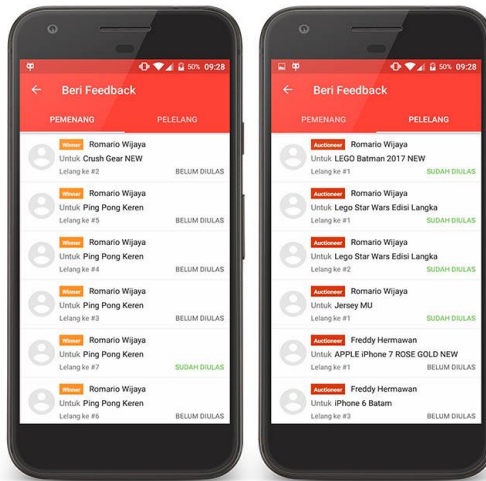
Kode Sumber 4.64 Menampilkan Muatan Halaman Beri *Feedback* untuk Pemenang

```

1.     <android.support.design.widget.CoordinatorLayout
2.         android:orientation="vertical">
3.         <include
4.             layout="@layout/toolbar_user_feedback" />
5.         <android.support.v4.view.ViewPager
6.             android:id="@+id/viewpager_berifedback" />
7.     </android.support.design.widget.CoordinatorLayout>

```

Kode Sumber 4.65 Menampilkan Halaman Beri *Feedback* untuk Pemenang



Gambar 4.8 Halaman *Beri Feedback* untuk Pelelang

4.3.9 Implementasi Fitur Detail *Beri Feedback*

Halaman *beri feedback* menampilkan detail dari ulasan yang dapat diberikan pada seorang pengguna. Jika ulasan belum pernah diberikan, maka pengguna dapat memasukkan ulasan baru. Jika ulasan sudah pernah diberikan sebelumnya, maka pengguna dapat memperbaharui ulasan yang sudah diberikan sebelumnya. Untuk implementasi muatan halaman detail *beri feedback* dapat dilihat pada Kode Sumber 4.66 dan implementasi antarmuka halaman detail *beri feedback* dapat dilihat pada Kode Sumber 4.67.

```

1. public DetailBeriFeedbackFragment()
2.     {
3.         initializeFragment(); //inisiasi fragment a
   nak.
4.     }
5.     @Override

```

```

6.     public void onCreate(Bundle savedInstanceState)
7.     {
8.         super.onCreate(savedInstanceState);
9.         getDataFromActivityResult(); //mendapatkan
           bundle data dari intent.
10.        initializeDataFeedbackReceiver(); //inisias
           i data receiver untuk menerima respon dari API keti
           ka memuat data feedback (jika feedback pernah diber
           ikan).
11.    }
12.    @Override
13.    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
14.    {
15.        View view = inflater.inflate(R.layout.fragment_user_berifedback_detail_layout, container, false);
16.        initializeViews(view); //memuat elemen-
           elemen view.
17.        setupViews(); // menampilkan indikator pera
           n pengguna yang akan diberikan ulasan (apakah pemen
           ang / pelelang)
18.        setupInitialFragment(); //memuat fragment y
           ang akan ditampilkan, tergantung apakah sebelumnya
           sudah pernah di ulas atau ulasan pertama.
19.        return view;
20.    }
21.    private void setupInitialFragment()
22.    {
23.        if (feedbackResources.isStatusRating()) { /
           /jika sudah pernah diulas
24.            setupFragment(loadingFeedbackFragment);
25.            getGivenFeedbackData(); //muat data ula
           san yang pernah diberikan sebelumnya.
26.        } else {
27.            setupFragment(giveFeedbackFragment);
28.        } }

```

Kode Sumber 4.66 Menampilkan Muatan Halaman Detail Beri *Feedback*

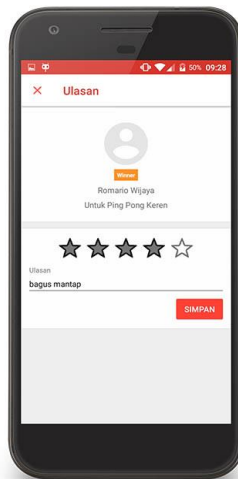
```

1. <ScrollView
2. xmlns:android="http://schemas.android.com/apk/res/a
   ndroid">
3. <LinearLayout android:orientation="vertical">
4. <View
5. android:layout_height="1dp"
6. android:background="@drawable/linear_horizontal_div
   ider"/>
7. <LinearLayout
8. android:orientation="vertical">
9. <ImageView
10. android:id="@+id/fragment_user_berifeedback_detail_
   avatar"/>
11. <TextView
12. android:id="@+id/fragment_user_berifeedback_detail_
   status_user"/>
13. <TextView
14. android:id="@+id/fragment_user_berifeedback_detail_
   nama_user"
15. android:text="Nama pengguna"/>
16. <LinearLayout
17. android:orientation="horizontal">
18. <TextView
19. android:text="Untuk "/>
20. <TextView
21. android:id="@+id/fragment_user_berifeedback_detail_
   nama_item"
22. android:text="Nama barang"/>
23. </LinearLayout>
24. </LinearLayout>
25. <View
26. android:layout_height="1dp"
27. android:background="@drawable/linear_horizontal_div
   ider"/>
28. <View
29. android:layout_height="1dp"
30. android:background="@drawable/linear_horizontal_div
   ider"/>
31. <FrameLayout
32. android:id="@+id/fragment_user_berifeedback_detail_
   layout">
33. </FrameLayout>
34. <View

```

```
35. android:layout_height="1dp"
36. android:background="@drawable/linear_horizontal_div
    ider"/>
37. </LinearLayout>
38. </ScrollView>
```

Kode Sumber 4.67 Menampilkan Halaman Detail Beri *Feedback*



Gambar 4.9 Halaman Detail Beri *Feedback*

4.3.10 Implementasi Fitur *Feedback* Anda

Halaman *feedback* anda menampilkan daftar ulasan yang diberikan pengguna lain terhadap pengguna yang telah melakukan *login* sebelumnya. Untuk implementasi muatan halaman *feedback* anda dapat dilihat pada Kode Sumber 4.68 dan implementasi antarmuka halaman *feedback* anda dapat dilihat pada Kode Sumber 4.69.

```

1.  @Override
2.      protected void onCreate(Bundle savedInstanceState)
3.      {
4.          super.onCreate(savedInstanceState);
5.          initializeFragments(); //inisiasi fragment a
6.          initializeViews(); //inisiasi view.
7.      }
8.      private void initializeViews()
9.      {
10.         ... //memuat elemen-elemen view
11.         setupViewPager(); //inisiasi viewpager
12.         tabLayout_feedbackAnda.setupWithViewPager(viewPager_feedbackAnda); //memasang viewpager dengan
13.         tablayout
14.     }
15.     private void initializeFragments()
16.     {
17.         winnerFragment = new WinnerFragment();
18.         auctioneerFragment = new AuctioneerFragment(
19.     );
20.     }
21.     private void setupViewPager()
22.     {
23.         FeedbackViewPagerAdapter pagerAdapter = new
24.         FeedbackViewPagerAdapter(getSupportFragmentManager(
25.     ));
26.         pagerAdapter.addFragment(winnerFragment, "Pe
27.         menang");
28.         pagerAdapter.addFragment(auctioneerFragment,
29.         "Pelelang");

```



```

24.     viewPager_feedbackAnda.setAdapter(pagerAdapt
25.     er);

```

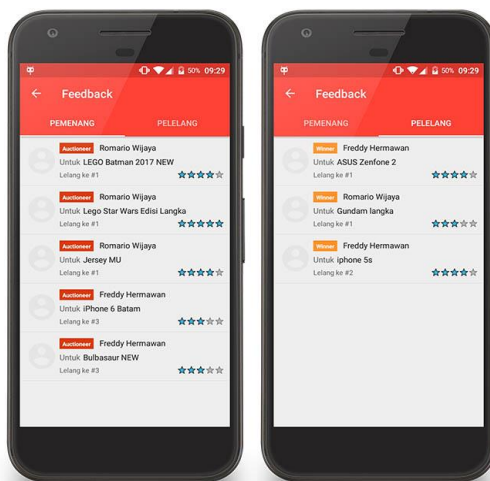
Kode Sumber 4.68 Menampilkan Muatan Halaman *Feedback* Anda

```

1. <android.support.design.widget.CoordinatorLayout
2. android:orientation="vertical">
3. <include
4. layout="@layout/toolbar_user_feedback" />
5. <android.support.v4.view.ViewPager
6. android:id="@+id/viewpager_feedbackanda"/>
7. </android.support.design.widget.CoordinatorLayout>

```

Kode Sumber 4.69 Menampilkan Halaman *Feedback* Anda



Gambar 4.10 Halaman *Feedback* Anda

4.3.11 Implementasi Fitur Riwayat

Halaman riwayat digunakan untuk menampilkan seluruh riwayat penawaran yang dilakukan oleh pengguna yang telah *login* sebelumnya. Untuk implementasi muatan halaman riwayat dapat dilihat pada Kode Sumber 4.70 dan implementasi antarmuka halaman riwayat dapat dilihat pada Kode Sumber 4.71.

```

1.      @Override
2.      public void onCreate(Bundle savedInstanceState)
3.      {
4.          super.onCreate(savedInstanceState);
5.          initializeConstant();
6.          initializeSession(); //mendapatkan data ses
   sion pengguna
7.          initializeChildFragments(); //inisiasi frag
   ment anak.
8.          setDataReceiverForRiwayat(); //inisiasi dat
   a receiver ketika data riwayat telah selesai dimasu
   kkan ArrayList.
9.      }
10.     @Override
11.     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
12.     {
13.         View view = inflater.inflate(R.layout.fragment_user_riwayat_layout, container, false);
14.         initializeViews(view);
15.         setSwipeRefreshLayoutProperties(); //member
   ikan properti kepada SwipeRefreshLayout untuk memua
   t data dari API.
16.         initializeWhenDataAlreadyReceived(); //inis
   iasi data receiver untuk menerima respon API yang b
   erisi daftar riwayat.
17.         return view;
18.     }
19.     private void setSwipeRefreshLayoutProperties()
20.     {

```

```

21.         swipeRefreshLayout.setColorSchemeResources(
22.             android.R.color.holo_orange_dark);
23.         swipeRefreshLayout.setOnRefreshListener(new
24.             SwipeRefreshLayout.OnRefreshListener() {
25.                 @Override
26.                 public void onRefresh() {
27.                     getRiwayat(userID);
28.                 }
29.             });
30.         swipeRefreshLayout.post(new Runnable() {
31.             @Override
32.             public void run() {
33.                 swipeRefreshLayout.setRefreshing(true);
34.                 getRiwayat(userID);
35.             }
36.         });

```

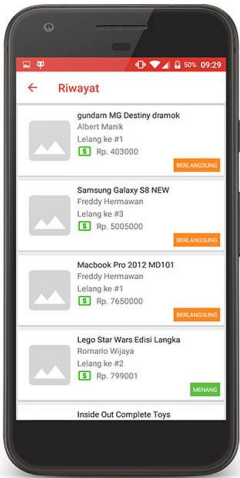
Kode Sumber 4.70 Menampilkan Muatan Halaman Riwayat

```

1. <android.support.v4.widget.SwipeRefreshLayout
2.     android:orientation="vertical"
3.     android:id="@+id/fragment_user_riwayat_swipeRefresh
4.     Layout">
5.     <FrameLayout
6.         android:id="@+id/fragment_user_riwayat_layout">
7.     </FrameLayout>
8. </android.support.v4.widget.SwipeRefreshLayout>

```

Kode Sumber 4.71 Menampilkan Halaman Riwayat



Gambar 4.11 Halaman Riwayat

4.3.12 Implementasi Fitur Detail Riwayat

Halaman detail riwayat digunakan untuk menampilkan seluruh riwayat harga penawaran pada sebuah sesi lelang yang telah dilakukan oleh pengguna yang telah *login* sebelumnya. Untuk implementasi muatan halaman detail riwayat dapat dilihat pada Kode Sumber 4.72 dan implementasi antarmuka halaman detail riwayat dapat dilihat pada Kode Sumber 4.73.

```

1. public DetailRiwayatFragment()
2. {
3.     initializeConstants();
4.     initializeFragment();
5.     initializeListDataReceiver();
6. }
7. @Override
8. public View onCreateView(LayoutInflater inflater, V
   viewGroup container, Bundle savedInstanceState)
9. {
10.     View view = inflater.inflate(R.layout.fragment_
      user_riwayat_detail_layout, container, false);
11.     initializeViews(view);
12.     initializeEssentialInformation(); //mengambil d
      ata dari intent dan menampilkannya.
13.     setupFragment/loadingListFragment); //menampilk
      an fragment awal, yaitu menampilkan loading bar.
14.     getBiddingList(); //meminta daftar tawaran dari
      API.
15.     return view;
16. }
17. private void setAndShowTheList()
18. {
19.     listNoEmptyFragment.setBiddingList(biddingList)
      ; //menampilkan daftar tawaran
20.     setupFragment(listNoEmptyFragment);
21. }
22. private void initializeListDataReceiver()
23. {
24.     listReceiver = new DataReceiver() {
25.         @Override
26.         public void dataReceived(Object output) {

```

```

27.         String response = output.toString();
28.         ... //mengambil data hasil respon API.

29.         setAndShowTheList(); //setelah selesai,
        panggil fungsi ini.
30.     }
31. };
32. }

```

Kode Sumber 4.72 Menampilkan Muatan Halaman Detail Riwayat

```

1. <LinearLayout
2.     android:orientation="vertical">
3.     <View
4.         android:layout_height="1dp"
5.         android:background="@drawable/linear_horizontal_div
        ider"/>
6.     <LinearLayout
7.         android:orientation="vertical">
8.         <TextView
9.             android:text="Informasi Item"/>
10.        <LinearLayout
11.            android:orientation="horizontal">
12.            <ImageView
13.                android:id="@+id/fragment_user_riwayat_detail_image
                "/>
14.            <LinearLayout
15.                android:orientation="vertical">
16.                <TextView
17.                    android:id="@+id/fragment_user_riwayat_detail_judul
                    item"
18.                    android:text="Judul Barang"/>
19.                <TextView
20.                    android:id="@+id/fragment_user_riwayat_detail_namaa
                    uctioneer"
21.                    android:text="Nama Auctioneer"/>
22.                <LinearLayout
23.                    android:orientation="horizontal">
24.                    <TextView
25.                        android:text="Lelang ke #"/>
26.                    <TextView

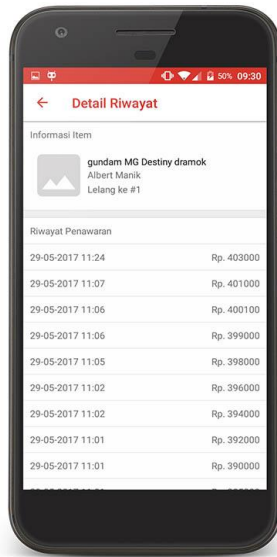
```

```

27. android:id="@+id/fragment_user_riwayat_detail_bidka
    li"/>
28. </LinearLayout> </LinearLayout> </LinearLayout>
29. </LinearLayout>
30. <View
31. android:layout_height="1dp"
32. android:background="@drawable/linear_horizontal_div
    ider"/>
33. <View
34. android:layout_height="1dp"
35. android:background="@drawable/linear_horizontal_div
    ider"/>
36. <LinearLayout>
37. <TextView
38. android:text="Riwayat Penawaran"/>
39. </LinearLayout>
40. <View
41. android:layout_height="1dp"
42. android:background="@drawable/linear_horizontal_div
    ider"/>
43. <FrameLayout
44. android:id="@+id/fragment_user_riwayat_detail_list_
    tawaran_layout">
45. </FrameLayout>
46. <View
47. android:layout_height="1dp"
48. android:background="@drawable/linear_horizontal_div
    ider"/>
49. </LinearLayout>

```

Kode Sumber 4.73 Menampilkan Halaman Detail Riwayat



Gambar 4.12 Halaman Detail Riwayat

4.3.13 Implementasi Fitur Favorit

Halaman favorit digunakan untuk melihat seluruh barang yang sudah dipilih oleh pengguna yang telah melakukan *login* sebelumnya sebagai favorit. Untuk implementasi muatan halaman favorit dapat dilihat pada Kode Sumber 4.74 dan implementasi antarmuka halaman favorit dapat dilihat pada Kode Sumber 4.75.

```

1. @Override
2. public void onCreate(Bundle savedInstanceState)
3. {
4.     super.onCreate(savedInstanceState);
5.     initializeSession(); //mendapatkan data session
   pengguna
6.     initializeChildFragment(); //menginisiasi fragm
   ent anak.
7. }
```



```

8. @Override
9. public View onCreateView(LayoutInflater inflater, V
   viewGroup container, Bundle savedInstanceState)
10. {
11.     View view = inflater.inflate(R.layout.fragment_
   user_favorite_layout, container, false);
12.     initializeViews(view); //memuat element view.
13.     initializeWhenDataAlreadyReceived(); //inisiasi
   data receiver ketika respon API diterima berisi da
   ftar favorit.
14.     initializeWhenListFavoriteIsEmptyReceiver(); //
   inisiasi data receiver ketika seluruh barang favori
   t dibuang.
15.     setSwipeRefreshLayoutProperties(); //memasang p
   roperti pada SwipeRefreshLayout untuk memuat API.
16.     return view;
17. }
18. private void setSwipeRefreshLayoutProperties()
19. {
20.     swipeRefreshLayout.setColorSchemeResources(android.R.color.holo_orange_dark);
21.     swipeRefreshLayout.setOnRefreshListener(new Swi
   peRefreshLayout.OnRefreshListener() {
22.         @Override
23.         public void onRefresh() {
24.             getDetailItemFavorites(userID);
25.         }
26.     });
27.     swipeRefreshLayout.post(new Runnable() {
28.         @Override
29.         public void run() {
30.             swipeRefreshLayout.setRefreshing(true);
31.             getDetailItemFavorites(userID);
32.         }
33.     });
34. }

```

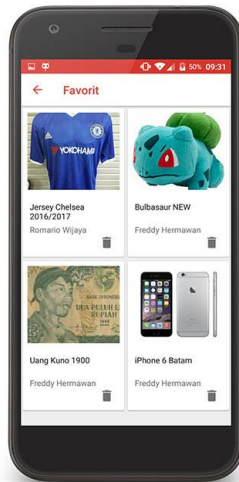
Kode Sumber 4.74 Menampilkan Muatan Halaman Favorit

```

1. <android.support.v4.widget.SwipeRefreshLayout
2.   android:orientation="vertical"
3.   android:id="@+id/fragment_user_favorite_swipe_refre
   shLayout">
4.   <FrameLayout
5.     android:id="@+id/fragment_user_favorite_layout">
6.   </FrameLayout>
7. </android.support.v4.widget.SwipeRefreshLayout>

```

Kode Sumber 4.75 Menampilkan Halaman Favorit



Gambar 4.13 Halaman Favorit

4.3.14 Implementasi Fitur Gerai

Halaman gerai digunakan untuk melihat seluruh barang yang dilelangkan oleh pengguna yang telah *login* sebelumnya. Untuk implementasi muatan halaman gerai dapat dilihat pada Kode Sumber 4.76 dan implementasi antarmuka halaman gerai dapat dilihat pada Kode Sumber 4.77.

```

1. @Override
2. public View onCreateView(LayoutInflater inflater, V
   iewGroup container, Bundle savedInstanceState){
3.     initializeSession(); //mengambil data session p
   engguna.
4.     view = inflater.inflate(R.layout.fragment_user_
   gerai_layout, container, false);
5.     initializeDataReceiver(); //inisiasi data recei
   ver untuk mengubah fragment ketika respon API sudah
   diterima.
6.     getBarangOnUserID(session.get(sessionManager.ge
   tKEY_ID())); //melakukan request ke API.
7.     return view;
8. }

```

Kode Sumber 4.76 Menampilkan Muatan Halaman Gerai

```

1. <!--fragment_user_gerei_layout-->
2. <FrameLayout>
3. <ProgressBar
4. android:id="@+id/fragment_user_gerei_layout_progres
   sbar" />
5. </FrameLayout>
6.
7. <!--fragment_user_gerei_layout_noempty-->
8. <android.support.v4.widget.SwipeRefreshLayout
9. android:id="@+id/fragment_user_gerei_layout_noempty
   _swipeRefreshLayout">
10. <android.support.v7.widget.RecyclerView
11. android:id="@+id/fragment_user_gerei_layout_recycle
   rview"
12. android:scrollbars="vertical" />
13. </android.support.v4.widget.SwipeRefreshLayout>

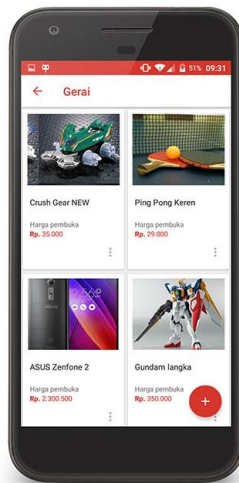
```

```

14. <!--fragment_user_gerai_layout_empty-->
15. <FrameLayout>
16. <LinearLayout
17.     android:orientation="vertical">
18.     <ImageView
19.         android:layout_width="96dp"
20.         android:layout_height="96dp"
21.         android:src="@drawable/ic_empty_basket_128dp"
22.         android:layout_gravity="center"/>
23.     <TextView
24.         android:text="@string/GERAI_EMPTY_MSGHEAD"/>
25.     <TextView
26.         android:text="@string/GERAI_EMPTY_MSGSUBHEAD"/>
27. </LinearLayout>
28. </FrameLayout>

```

Kode Sumber 4.77 Menampilkan Halaman Gerai



Gambar 4.14 Halaman Gerai

4.3.15 Implementasi Fitur Lelang Barang

Halaman lelang barang digunakan untuk mendaftarkan barang baru yang akan dilelang oleh pengguna yang telah melakukan *login* sebelumnya. Untuk implementasi muatan halaman lelang barang dapat dilihat pada Kode Sumber 4.78 dan implementasi antarmuka halaman lelang barang dapat dilihat pada Kode Sumber 4.79.

```

1. @Override
2. public View onCreateView(LayoutInflater inflater, V
   viewGroup container, Bundle savedInstanceState){
3.     View view = inflater.inflate(R.layout.fragment_
   user_lelang_barang_layout, container, false);
4.     initializeSession(); //mengambil data session p
   engguna
5.     initializeViews(view); //memuat elemen-
   elemen view
6.     initializeAlertDialogImageBuilder(); //inisiasi
   alert dialog untuk memasukkan gambar
7.     initializeTextChangeListenerOnPrice(); //inisia
   si listener ketika pengguna mengetik harga.
8.     initializeImageReceiver(); //inisiasi receiver
   ketika pengguna mengirim gambar ke server.
9.     setupAllInputValidation(); //memasang properti
   untuk validasi input pada EditText
10.    setImagePicker(); //inisiasi receiver ketika pe
   ngguna memilih gambar
11.    setupViews(); //memasang
   properti element view yang dapat di klik
12.    initializeAdapter(); //inisiasi adapter untuk d
   aftar gambar barang
13.    setRecyclerViewProperties(); //inisiasi properti
   i recyclerView untuk daftar gambar barang
14.    return view;
15. }

```

Kode Sumber 4.78 Menampilkan Muatan Halaman Lelang Barang

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <ScrollView android:id="@+id/fragment_user_lelang_b
   arang_scroll">
3. <LinearLayout
4.   android:orientation="vertical">
5.   <TextView
6.     android:text="Nama barang"/>
7.   <android.support.v7.widget.CardView>
8.     <android.support.design.widget.TextInputLayout
9.       android:id="@+id/fragment_user_lelang_barang_input_
       nama_barang">
10.      <EditText
11.        android:id="@+id/fragment_user_lelang_barang_nama_b
        arang" />
12.      </android.support.design.widget.TextInputLayout>
13.    </android.support.v7.widget.CardView>
14.    <TextView
15.      android:text="Foto barang"/>
16.    <android.support.v7.widget.RecyclerView
17.      android:id="@+id/fragment_user_lelang_barang_image_
      recyclerview"
18.      android:orientation="horizontal">
19.    </android.support.v7.widget.RecyclerView>
20.    <TextView
21.      android:id="@+id/fragment_user_lelang_barang_image_
      warning"/>
22.    <TextView
23.      android:text="Kategori Barang"/>
24.    <android.support.v7.widget.CardView>
25.      <Spinner
26.        android:id="@+id/fragment_user_lelang_barang_katego
        ri"
27.        android:entries="@array/categories">
28.      </Spinner>
29.    </android.support.v7.widget.CardView>
30.    <TextView
31.      android:text="Detail barang"/>
32.    <android.support.v7.widget.CardView>
33.      <LinearLayout
34.        android:orientation="vertical">
35.      <android.support.design.widget.TextInputLayout
36.        android:id="@+id/fragment_user_lelang_barang_input_
        deskripsi_barang">

```

```

37. <EditText
38. android:id="@+id/fragment_user_lelang_barang_deskri
   psi_barang"
39. android:hint="Deskripsi barang"/>
40. </android.support.design.widget.TextInputLayout>
41. <android.support.design.widget.TextInputLayout
42. android:id="@+id/fragment_user_lelang_barang_input_
   harga_awal">
43. <EditText
44. android:id="@+id/fragment_user_lelang_barang_harga_
   awal_barang"
45. android:hint="Harga awal"/>
46. </android.support.design.widget.TextInputLayout>
47. <android.support.design.widget.TextInputLayout
48. android:id="@+id/fragment_user_lelang_barang_input_
   harga_target">
49. <EditText
50. android:id="@+id/fragment_user_lelang_barang_harga_
   target_barang"
51. android:hint="Harga target"/>
52. </android.support.design.widget.TextInputLayout>
53. </LinearLayout>
54. </android.support.v7.widget.CardView>
55. <TextView
56. android:text="Atur jadwal lelang"/>
57. <android.support.v7.widget.CardView>
58. <LinearLayout>
59. <TextView
60. android:id="@+id/fragment_user_lelang_barang_jadwal_
   _warning"/>
61. <TextView
62. android:text="Jadwal Mulai"/>
63. <LinearLayout
64. android:orientation="horizontal">
65. <TextView
66. android:id="@+id/fragment_user_lelang_barang_tangga
   l_mulai"/>
67. <TextView
68. android:id="@+id/fragment_user_lelang_barang_jam_mu
   lai"/>
69. </LinearLayout>
70. <TextView
71. android:text="Jadwal Selesai"/>

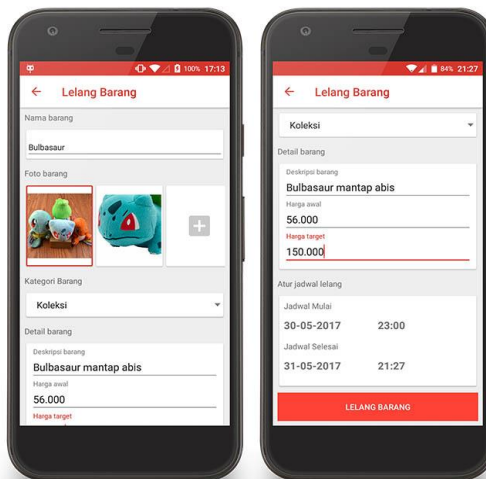
```

```

72. <LinearLayout
73.     android:orientation="horizontal">
74. <TextView
75.     android:id="@+id/fragment_user_lelang_barang_tangga
       l_selesai"/>
76. <TextView
77.     android:id="@+id/fragment_user_lelang_barang_jam_se
       lesai"/>
78. </LinearLayout>
79. </LinearLayout>
80. </android.support.v7.widget.CardView>
81. <Button
82.     android:id="@+id/fragment_user_lelang_barang_jual_b
       utton"
83.     android:text="Lelang Barang"/>
84. </LinearLayout>
85. </ScrollView>

```

Kode Sumber 4.79 Menampilkan Halaman Lelang Barang



Gambar 4.15 Halaman Lelang Barang

4.3.16 Implementasi Fitur *Edit Barang*

Halaman edit barang digunakan untuk mengubah informasi barang yang sudah terdaftar. *Edit* barang dapat dilakukan ketika barang masih belum mulai masuk masa lelang dan setelah lelang berakhir, dengan catatan pemenang lelang sudah terpilih. Ketika edit barang dilakukan saat barang masih belum mulai masuk masa lelang, maka edit hanya bertujuan untuk mengubah informasi semata. Namun jika dilakukan saat masa lelang sudah berakhir dan pemenang telah terpilih, maka edit barang bertujuan untuk memulai sesi lelang baru dengan barang yang sama. Maka, pengguna dapat melelangkan barang yang sama berkali-kali. Untuk implementasi muatan halaman edit barang dapat dilihat pada Kode Sumber 4.80 dan implementasi antarmuka halaman edit barang dapat dilihat pada Kode Sumber 4.81.

```

1. @Override
2. public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState){
3.     View view = inflater.inflate(R.layout.fragment_user_edit_lelang_barang_layout, container, false);
4.     initializeViews(view); //memuat elemen-elemen view.
5.     initializeTextChangedListenerOnPrice(); //inisiasi listener ketika pengguna memasukkan harga
6.     initializeDateTimeOnClickListeners(); //memasang properti elemen view yang dapat di klik untuk memunculkan tampilan tanggal dan waktu.
7.     setupAllInputValidation(); //memasang properti validasi input pada EditText
8.     takeMainImageAsFirstImage(); //memindahkan gambar utama ke urutan awal.
9.     setInitialUniqueIdImage(); //mencatat ID dari gambar utama.
10.    setImagePicker(); //inisiasi receiver ketika pengguna memilih gambar.
11.    setDataBarangInfo(); //menampilkan data barang yang telah diterima di elemen-elemen view.

```

```

12.     initializeAdapter(); //inisiasi adapter untuk d
    aftar gambar.
13.     initializeImageReceiver(); //inisiasi data rece
    iver ketika pengguna mengirim pembaharuan informasi
    gambar ke server.
14.     setImageLoaderReceiver(); //inisiasi receiver u
    ntuk menerima data bitmap gambar-
    gambar dari server
15.     setupRecyclerViewProperties(); //inisiasi recyc
    lerview untuk daftar gambar
16.     loadExistingImage(); //mengambil data gambar da
    ri server
17.     setViewClickListener(); //memasang properti ele
    men-view yang dapat di klik
18.     return view;
19. }

```

Kode Sumber 4.80 Menampilkan Muatan Halaman *Edit* Barang

```

1. <ScrollView
2.     android:id="@+id/fragment_user_edit_lelang_barang_s
    croll">
3.     <LinearLayout
4.         android:orientation="vertical">
5.         <TextView
6.             android:text="Nama barang"/>
7.         <android.support.v7.widget.CardView>
8.             <android.support.design.widget.TextInputLayout
9.                 android:id="@+id/fragment_user_edit_lelang_barang_i
    nput_nama_barang">
10.                 <EditText
11.                     android:id="@+id/fragment_user_edit_lelang_barang_n
    ama_barang" />
12.                 </android.support.design.widget.TextInputLayout>
13.             </android.support.v7.widget.CardView>
14.             <TextView
15.                 android:text="Foto barang (maksimal 8 gambar)"/>
16.             <android.support.v7.widget.RecyclerView
17.                 android:id="@+id/fragment_user_edit_lelang_barang_i
    mage_recyclerview">
18.             </android.support.v7.widget.RecyclerView>
19.             <TextView

```

```

20. android:id="@+id/fragment_user_edit_lelang_barang_i
    mage_warning"/>
21. <TextView
22. android:text="Kategori Barang"/>
23. <android.support.v7.widget.CardView>
24. <Spinner
25. android:id="@+id/fragment_user_edit_lelang_barang_k
    ategori"
26. android:entries="@array/categories">
27. </Spinner>
28. </android.support.v7.widget.CardView>
29. <TextView
30. android:text="Detail barang"/>
31. <android.support.v7.widget.CardView>
32. <LinearLayout
33. android:orientation="vertical">
34. <android.support.design.widget.TextInputLayout
35. android:id="@+id/fragment_user_edit_lelang_barang_i
    nput_deskripsi_barang">
36. <EditText
37. android:id="@+id/fragment_user_edit_lelang_barang_d
    eskripsi_barang"
38. android:hint="Deskripsi barang"/>
39. </android.support.design.widget.TextInputLayout>
40. <android.support.design.widget.TextInputLayout
41. android:id="@+id/fragment_user_edit_lelang_barang_i
    nput_harga_awal">
42. <EditText
43. android:id="@+id/fragment_user_edit_lelang_barang_h
    arga_awal_barang"
44. android:hint="Harga awal"/>
45. </android.support.design.widget.TextInputLayout>
46. <android.support.design.widget.TextInputLayout
47. android:id="@+id/fragment_user_edit_lelang_barang_i
    nput_harga_target">
48. <EditText
49. android:id="@+id/fragment_user_edit_lelang_barang_h
    arga_target_barang"
50. android:hint="Harga target"/>
51. </android.support.design.widget.TextInputLayout>
52. </LinearLayout>
53. </android.support.v7.widget.CardView>
54. <TextView

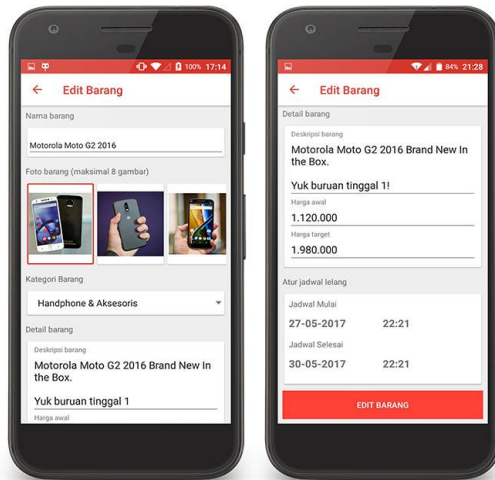
```

```

55. android:text="Atur jadwal lelang"/>
56. <android.support.v7.widget.CardView>
57. <LinearLayout
58. android:orientation="vertical">
59. <TextView
60. android:id="@+id/fragment_user_edit_lelang_barang_j
    adwal_warning"/>
61. <TextView
62. android:text="Jadwal Mulai"/>
63. <LinearLayout
64. android:orientation="horizontal">
65. <TextView
66. android:id="@+id/fragment_user_edit_lelang_barang_t
    anggal_mulai"/>
67. <TextView
68. android:id="@+id/fragment_user_edit_lelang_barang_j
    am_mulai"/>
69. </LinearLayout>
70. <TextView
71. android:text="Jadwal Selesai"/>
72. <LinearLayout
73. android:orientation="horizontal">
74. <TextView
75. android:id="@+id/fragment_user_edit_lelang_barang_t
    anggal_selesai"/>
76. <TextView
77. android:id="@+id/fragment_user_edit_lelang_barang_j
    am_selesai"/>
78. </LinearLayout>
79. </LinearLayout>
80. </android.support.v7.widget.CardView>
81. <Button
82. android:id="@+id/fragment_user_edit_lelang_barang_j
    ual_button"
83. android:text="Edit Barang"/>
84. </LinearLayout>
85. </ScrollView>

```

Kode Sumber 4.81 Menampilkan Halaman *Edit* Barang



Gambar 4.16 Halaman *Edit Barang*

4.3.17 Implementasi Fitur Detail Barang

Halaman detail barang berguna untuk menampilkan berbagai informasi mengenai sebuah barang. Terdapat dua versi halaman detail barang pada perangkat lunak Lelangapa, yaitu versi untuk pengguna umum (selanjutnya disebut sebagai versi I) dan versi untuk pengguna sebagai pelelang barang yang bersangkutan (selanjutnya disebut sebagai versi II).

Halaman detail barang versi I memungkinkan pengguna untuk memasukkan harga tawaran yang ingin diajukan, melihat deskripsi barang, melihat tiga besar peringkat pengguna dengan tawaran tertinggi, menjadikan barang sebagai barang favorit, akses untuk menghubungi pelelang melalui pesan singkat, dan melihat foto-foto barang yang dilelangkan. Untuk implementasi muatan halaman detail barang versi I dapat dilihat pada Kode Sumber 4.82..

Halaman detail barang versi II memungkinkan pelemang untuk melihat aktivitas lelang, akses untuk melihat daftar tawaran yang masuk, melihat statistik lelang, melihat deskripsi lelang, dan akses untuk melakukan edit barang. Untuk implementasi muatan halaman detail barang versi II dapat dilihat pada Kode Sumber 4.84. Sementara implementasi antarmuka halaman detail barang versi I dan II dapat dilihat pada Kode Sumber 4.83

```

1. @Override
2. public void onCreate(Bundle savedInstanceState)
3. {
4.     super.onCreate(savedInstanceState);
5.     setHasOptionsMenu(true);
6.
7.     initializeIntentToChatActivity(); //inisiasi in
tent ke ChatActivity
8.     initializeSession(); //mengambil data session p
engguna
9.     getIntentData(); //mendapatkan data dari bundle
intent
10.    initializeAlertDialogBuilder(); //inisiasi aler
t dialog builder
11.    setInputBidReceiver(); //inisiasi receiver untu
k membentuk data dikirim ke socket
12.    setSocketReceiver(); //inisiasi receiver ketika
ada data masuk dari socket
13.    setInitialSocketBinding(); //inisiasi socket pe
nawaran
14.    setDetailReceived(); //inisiasi data receiver s
aat data detail penawaran diterima dari respon API.
15.    setTriggerReceived(); //inisiasi data receiver
untuk memicu alert dialog saat lelang mulai/selesai
16.    setDataReceiverFavoriteStatus(); //inisiasi dat
a receiver untuk respon API favorit
17.    initializeConstants(); //inisiasi objek
18.    initializeFragments(); //inisiasi fragment anak
19. }
20. @Override
21. public View onCreateView (final LayoutInflater infl
ater, ViewGroup container, Bundle savedInstanceState)

```

```

22. {
23.     View view = inflater.inflate(R.layout.fragment_
        detail_barang_layout, container, false);
24.     initializeViews(view);
25.     setUpSwipeRefreshLayout(); //memasang properti
        SwipeRefreshLayout sehingga dapat memuat ulang API
        return view;
26. }

```

Kode Sumber 4.82 Menampilkan Muatan Halaman Detail Barang versi I

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <android.support.v4.widget.SwipeRefreshLayout
3. android:id="@+id/fragment_detail_barang_swipe_refre
    shLayout">
4. <ScrollView>
5. <LinearLayout
6. android:orientation="vertical">
7. <FrameLayout
8. android:id="@+id/fragment_detail_barang_header_frag
    ment">
9. </FrameLayout>
10. <FrameLayout
11. android:id="@+id/fragment_detail_barang_gambar_frag
    ment">
12. </FrameLayout>
13. <FrameLayout
14. android:id="@+id/fragment_detail_barang_waktubid_fr
    agment">
15. </FrameLayout>
16. <FrameLayout
17. android:id="@+id/fragment_detail_barang_auctioneer_
    fragment">
18. </FrameLayout>
19. <FrameLayout
20. android:id="@+id/fragment_detail_barang_bidding_fra
    gment">
21. </FrameLayout>
22. <FrameLayout
23. android:id="@+id/fragment_detail_barang_deskripsi_f
    ragment">
24. </FrameLayout>

```

```

25. <FrameLayout
26.   android:id="@+id/fragment_detail_barang_komentar_fr
    agment">
27. </FrameLayout>
28. </LinearLayout>
29. </ScrollView>
30. </android.support.v4.widget.SwipeRefreshLayout>

```

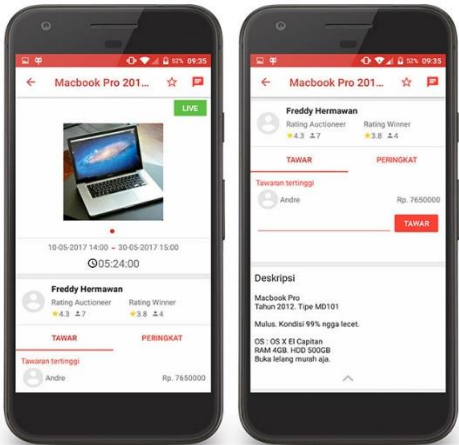
Kode Sumber 4.83 Menampilkan Halaman Detail Barang versi I dan II

```

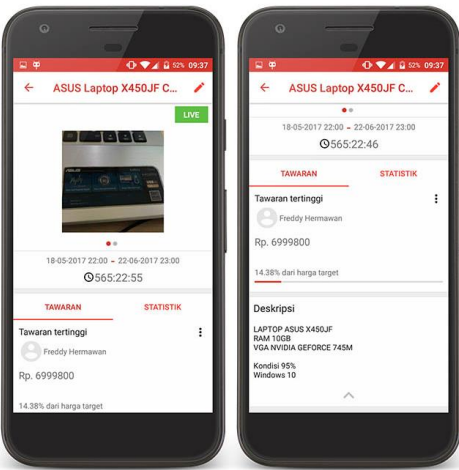
1. @Override
2. public void onCreate(Bundle savedInstanceState)
3. {
4.     super.onCreate(savedInstanceState);
5.     setHasOptionsMenu(true);
6.     initializeConstants(); //inisiasi objek
7.     createAlertDialogCannotEditBuilder(); //inisias
    i alert dialog untuk pemberitahuan tidak bisa melak
    ukan pembaharuan data barang.
8.     initializeSocketReceiver(); //inisiasi receiver
    ketika ada data dari socket masuk
9.     initializeSocket(); //inisiasi socket
10.    initializeDataReceiver(); //inisiasi data recei
    ver yang digunakan
11. }
12. @Override
13. public View onCreateView(LayoutInflater inflater, V
    iewGroup container, Bundle savedInstanceState)
14. {
15.     View view = inflater.inflate(R.layout.fragment_
    detail_barang_layout, container, false);
16.     initializeViews(view); //memuat elemen-
    elemen view
17.     setUpSwipeRefreshLayout(); //memasang properti
    pada SwipeRefreshLayout untuk memuat data dari API
18.     return view;
19. }

```

Kode Sumber 4.84 Menampilkan Muatan Halaman Detail Barang versi II



Gambar 4.17 Halaman Detail Barang versi I



Gambar 4.18 Halaman Detail Barang versi II

4.3.18 Implementasi Fitur Pencarian

Halaman pencarian digunakan untuk melakukan pencarian barang. Pengguna dapat mengetikkan kata kunci pencarian dan perangkat lunak akan menampilkan daftar barang yang relevan dengan kata kunci yang telah dimasukkan pengguna. Jika tidak terdapat barang yang sesuai dengan kata kunci, maka perangkat lunak akan menampilkan pemberitahuan bahwa barang tidak tersedia. Untuk implementasi muatan halaman pencarian dapat dilihat pada Kode Sumber 4.86 dan implementasi antarmuka halaman pencarian dapat dilihat pada Kode Sumber 4.87.

```

1. @Override
2. protected void onCreate(Bundle savedInstanceState){
3.     super.onCreate(savedInstanceState);
4.     initializeConstants(); //inisiasi objek
5.     initializeFilterManager(); //inisiasi penyaring
   an pencarian
6.     initializeViews(); //memuat elemen-
   elemen view.
7.     initializeFragments(); //inisiasi fragment anak
8.     setupStarterFragment(); //pasang fragment awal
9. }
10. @Override
11. public boolean onCreateOptionsMenu(Menu menu){
12.     MenuInflater inflater = getMenuInflater();
13.     inflater.inflate(R.menu.activity_main_searc
   h_menu, menu);
14.     setupSearchViewProperties();
15.     searchView.setOnQueryTextListener(new SearchView
   .OnQueryTextListener() {
16.         //mengatur pergantian fragment ketika ada k
   ata kunci yang diketik
17.         @Override
18.         public boolean onQueryTextSubmit(String que
   ry) {
19.             if (!switchToTextQuerySubmitFragment){
20.                 currentFragment = getFragmentManager
   r().findFragmentByTag("FRAGMENT_TEXT_SUBMIT");
21.                 if (currentFragment==null)
22.                     {

```

```

23.                ...//ubah ke fragment TEXT_SUBMIT
24.            }
25.        }
26.        return false;
27.    }
28.
29.    @Override
30.    public boolean onQueryTextChange(String new
Text) {
31.        if (newText.trim().isEmpty()) {
32.            ...//set back to START FRAGMENT
33.        }
34.        else {
35.            if (!switchToTextChangeFragment){
36.                currentFragment = getFragmentManager().findFragmentByTag("FRAGMENT_TEXT_CHANGE");
37.                if (currentFragment==null)
38.                {
39.                    ...//ubah ke fragment TEXT CHANGE
40.                }
41.            }
42.            else {
43.                textChangeFragment.changeText(newText);
44.            }
45.        }
46.        return false; }
47.    });
48.    return super.onCreateOptionsMenu(menu); }

```

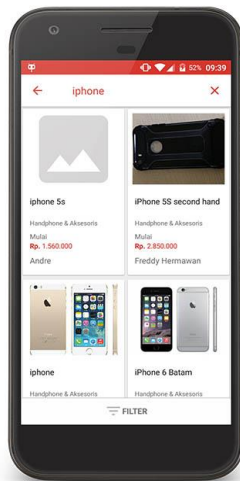
Kode Sumber 4.85 Menampilkan Muatan Halaman Pencarian

```

1. <android.support.design.widget.CoordinatorLayout
2. android:id="@+id/activity_main_search_coordinatorLayout">
3. <include
4. layout="@layout/toolbar" />
5. <FrameLayout
6. android:id="@+id/fragment_main_search_textchange">
7. </FrameLayout>
8. </android.support.design.widget.CoordinatorLayout>

```

Kode Sumber 4.86 Menampilkan Halaman Pencarian



Gambar 4.19 Halaman Pencarian

4.3.19 Implementasi Fitur Filter Pencarian

Halaman *filter* pencarian digunakan untuk menyaring hasil pencarian yang sudah dilakukan berdasarkan parameter tertentu. Parameter yang digunakan pada penyaringan pencarian saat ini adalah *filter* pencarian berdasarkan kategori. Halaman *filter* pencarian hanya dapat diakses jika hasil pencarian berdasarkan kata kunci ditemukan. Untuk implementasi muatan halaman penyaringan pencarian dapat dilihat pada Kode Sumber 4.88 dan antarmuka halaman penyaringan pencarian dapat dilihat pada Kode Sumber 4.89.

```

1. @Override
2. public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
3.     View view = inflater.inflate(R.layout.fragment_filter_layout, container, false);
4.     initializeConstants(); //inisiasi objek

```

```

5.     initializeViews(view); //memuat elemen-
        elemen view
6.     initializeAdapters(); //inisiasi adapter, dan m
        emasukkan elemen filter ke dalam arraylist
7.     setRecyclerViewProperties(); //memasang propert
        i pada recyclerview untuk daftar filter
8.     buildCategoryDialog(); //inisiasi dialog untuk
        memunculkan daftar kategori
9.     initializeClickListeners(); //memasang properti
        elemen view yang dapat di klik
10.    return view;
11. }
12. private void buildCategoryDialog() {
13.     categoryAlertDialogBuilder = new AlertDialog.Bu
        ilder(getActivity());
14.     categoryAlertDialogBuilder.setTitle("Kategori")

15.         .setItems(listAllCategory, new DialogIn
        terface.OnClickListener() {
16.             @Override
17.             public void onClick(DialogInterface
        dialog, int which) {
18.                 whenCategoryChosen(which);
19.             }
20.         });
21. }
22. private void whenCategoryChosen(int idx) {
23.     FilterManager.insertFilter(Integer.toString(idx
        ));
24.     filter.get(0).setValueFilter(listAllCategory[id
        x]);
25.     adapter.notifyDataSetChanged();
26. }
27. private void initializeClickListeners() {
28.     button_applyfilter.setOnClickListener(new View.
        OnClickListener() {
29.         @Override
30.         public void onClick(View v) {
31.             //susun jsonarray berdasarkan filter ya
        ng sudah dipilih
32.             if (!FilterManager.getFilter().get(Filt
        erManager.KEY_ID_CATEGORY).equals("0")) FilterManag
        er.IS_FILTER_SPECIFIED = true;

```

```

33.         else FilterManager.IS_FILTER_SPECIFIED
34.         = false;
35.         buildParamsJSONObjectAndSendResult();
36.     }
37. });
38. }

```

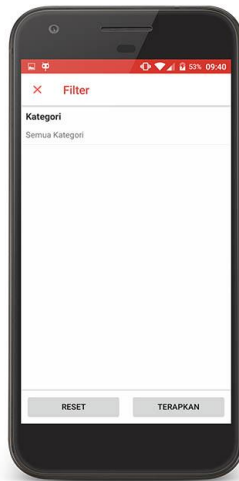
Kode Sumber 4.87 Menampilkan Muatan Halaman Penyaringan Pencarian

```

1. <LinearLayout
2.     android:orientation="vertical">
3.     <android.support.v7.widget.RecyclerView
4.         android:id="@+id/fragment_filter_layout_recyclervie
5.         w"
6.         android:scrollbars="vertical" />
7.     <LinearLayout
8.         android:orientation="horizontal">
9.         <Button
10.            android:id="@+id/fragment_filter_layout_button_rese
11.            t"
12.            android:text="RESET"/>
13.         </FrameLayout>
14.         <FrameLayout>
15.             <Button
16.                 android:id="@+id/fragment_filter_layout_button_filt
17.                 er"
18.                 android:text="TERAPKAN"/>
19.             </FrameLayout>
20.         </LinearLayout>
21.     </LinearLayout>

```

Kode Sumber 4.88 Menampilkan Halaman Penyaringan Pencarian



Gambar 4.20 Halaman Penyaringan Pencarian

4.3.20 Implementasi Fitur Kategori

Halaman kategori digunakan untuk menampilkan semua barang yang merupakan bagian dari kategori tertentu. Untuk implementasi muatan halaman kategori dapat dilihat pada Kode Sumber 4.90 dan implementasi antarmuka halaman kategori dapat dilihat pada Kode Sumber 4.91.

```

1. public ItemCategoryFragment() {
2.     initializeConstants(); //inisiasi objek
3. }
4. @Override
5. public void onCreate(Bundle savedInstanceState) {
6.     super.onCreate(savedInstanceState);
7.     getIntentData(); //mendapatkan data dari intent
8.     initializeFragments(); //inisiasi fragment anak

```

```

9.     initializeDataReceiver(); //inisiasi objek data
    receiver
10. }
11. @Override
12. public View onCreateView(LayoutInflater inflater, V
    iewGroup container, Bundle savedInstanceState) {
13.     View view = inflater.inflate(R.layout.fragment_
    item_category_layout, container, false);
14.     initializeViews(view); //memuat elemen-
    elemen view
15.     setSwipeRefreshLayoutProperties(); //memasang p
    roprieti SwipeRefreshLayout untuk memuat API
16.     return view;
17. }
18. private void initializeDataReceiver() {
19.     whenItemListLoaded = new DataReceiver() {
20.         @Override
21.         public void dataReceived(Object output) {
22.             String response = output.toString();
23.             ...//memasukkan seluruh data barang
    yang diperoleh dari respon API ke ArrayList
24.             whenAllDataLoaded();
25.         } catch (JSONException e) {
26.             e.printStackTrace();
27.         }
28.     }
29. };
30. }
31. private void whenAllDataLoaded() {
32.     if (listItems.isEmpty()) {
33.         //jika ArrayList kosong, tampilkan emptyFra
    gment
34.         setupFragment(emptyFragment);
35.     }
36.     else {
37.         //jika ada isinya, tampilkan noEmptyFragmen
    t
38.         setupFragment(noEmptyFragment);
39.     }
40. }

```

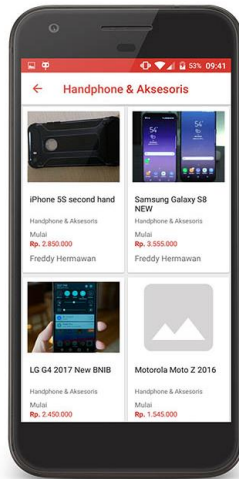
Kode Sumber 4.89 Menampilkan Muatan Halaman Kategori


```

1. <android.support.v4.widget.SwipeRefreshLayout
2.   android:id="@+id/fragment_item_category_layout_swip
   eRefreshLayout">
3.   <FrameLayout
4.     android:id="@+id/fragment_item_category_layout">
5.   </FrameLayout>
6. </android.support.v4.widget.SwipeRefreshLayout>
7.
8. <!--fragment_category_empty-->
9. <FrameLayout>
10. <LinearLayout
11.   android:orientation="vertical">
12.   <ImageView
13.     android:src="@drawable/ic_search_grey_128dp"/>
14.   <TextView
15.     android:text="Pencarian tidak ditemukan"/>
16. </LinearLayout>
17. </FrameLayout>
18.
19. <!--fragment_category_noempty-->
20. <LinearLayout
21.   android:orientation="vertical">
22.   <android.support.v7.widget.RecyclerView
23.     android:id="@+id/fragment_item_category_layout_recy
       clerview"
24.     android:scrollbars="vertical" />
25. </LinearLayout>

```

Kode Sumber 4.90 Menampilkan Halaman Kategori



Gambar 4.21 Halaman Kategori

4.3.21 Implementasi Fitur Daftar *Chat*

Halaman daftar *chat* atau daftar pesan singkat digunakan untuk melihat seluruh daftar pesan singkat yang dikirim oleh satu pengguna, yaitu pengguna yang telah melakukan *login* sebelumnya, ke pengguna lainnya. Untuk implementasi muatan halaman daftar chat dapat dilihat pada Kode Sumber 4.92 dan implementasi antarmuka halaman daftar chat dapat dilihat pada Kode Sumber 4.93.

```

1. @Override
2. public void onCreate(Bundle savedInstanceState)
3. {
4.     super.onCreate(savedInstanceState);
5.     initializeChildFragments(); //inisiasi fragment
    anak
6.     initializeDataReceiver(); //inisiasi data receiver
    untuk respon API
7.     initializeSession(); //mendapatkan data session
    pengguna

```

```

8.  }
9.  @Override
10. public View onCreateView(LayoutInflater inflater, V
    iewGroup container, Bundle savedInstanceState)
11. {
12.     View view = inflater.inflate(R.layout.fragment_
    chat_room_list_layout, container, false);
13.     initializeViews(view); //memuat elemen elemen v
    iew
14.     setSwipeRefreshLayoutProperties(); //memasang p
    roprieti SwipeRefreshLayout untuk memuat API
15.     return view;
16. }
17. private void initializeDataReceiver()
18. {
19.     dataReceiver = new DataReceiver() {
20.         @Override
21.         public void dataReceived(Object output) {
22.             String response = output.toString();
23.             ...//masukkan semua daftar ruangan
    kedalam ArrayList
24.             whenDataChatRoomsAlreadyReceived();
25.         } catch (JSONException e) {
26.             e.printStackTrace();
27.         }
28.     }
29. };
30. }
31. private void whenDataChatRoomsAlreadyReceived()
32. {
33.     if (listChatRoom.isEmpty()) {
34.         ...//jika arraylist kosong, jalankan emptyF
    ragment
35.     }
36.     else {
37.         ...//jika arraylist ada isinya, jalankan no
    emptyFragment
38.     }
39. }

```

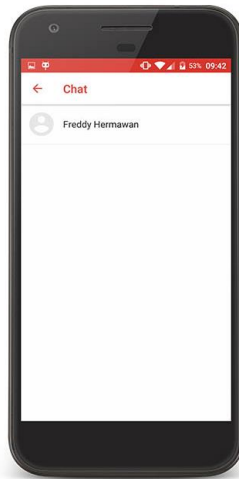
Kode Sumber 4.91 Menampilkan Muatan Halaman Daftar Chat

```

1. <android.support.v4.widget.SwipeRefreshLayout
2.   android:id="@+id/fragment_chat_room_list_layout_swipe_refreshLayout">
3.   <FrameLayout
4.     android:id="@+id/fragment_chat_room_list_layout">
5.   </FrameLayout>
6. </android.support.v4.widget.SwipeRefreshLayout>
7.
8. <!--fragment_chat_list_empty-->
9. <FrameLayout>
10. <LinearLayout
11.   android:orientation="vertical">
12.   <ImageView
13.     android:src="@drawable/ic_chat_empty_128dp"/>
14.   <TextView
15.     android:text="Tidak ada pesan" />
16. </LinearLayout>
17. </FrameLayout>
18.
19. <!--fragment_chat_list_noempty-->
20. <android.support.v4.widget.SwipeRefreshLayout
21.   android:id="@+id/fragment_chat_room_list_layout_noempty_swipeRefreshLayout">
22.   <android.support.v7.widget.RecyclerView
23.     android:id="@+id/fragment_chat_room_list_layout_recyclerview"
24.     android:scrollbars="vertical"/>
25. </android.support.v4.widget.SwipeRefreshLayout>

```

Kode Sumber 4.92 Menampilkan Halaman Daftar Chat



Gambar 4.22 Halaman Daftar Chat

4.3.22 Implementasi Fitur Ruang *Chat*

Halaman ruang *chat* atau ruang pesan singkat digunakan untuk berkirim pesan antara satu pengguna ke satu pengguna lainnya. Untuk implementasi muatan halaman ruang chat dapat dilihat pada Kode Sumber 4.94 dan implementasi antarmuka halaman ruang chat dapat dilihat pada Kode Sumber 4.95.

```

1. @Override
2. public void onCreate(Bundle savedInstanceState)
3. {
4.     super.onCreate(savedInstanceState);
5.     initializeConstants(); //inisiasi objek
6.     initializeSocketReceivers(); //inisiasi receiver untuk menerima data yang masuk dari socket
7.     initializeSocket(); //inisiasi socket
8. }
9. @Override
10. public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

```

```

11. {
12.     View view = inflater.inflate(R.layout.fragment_
chat_room_message_layout, container, false);
13.     initializeViews(view); //memuat elemen elemen v
iew
14.     checkViewScreenSize(view); //digunakan untuk me
lakukan scroll ke bawah otomatis ketika keyboard ak
tif
15.     setOnClickListener(); //inisiasi objek onClickL
istener untuk digunakan oleh elemen view.
16.     setViewProperties(); //memasang properti eleme
n view yang dapat di klik
17.     setRecyclerViewAdapter(); //inisiasi adapter pe
san
18.     setRecyclerViewProperties(); //memasang propert
i recyclerview
19.     connectSocket(); //menghubungkan socket
20.     return view;
21. }

```

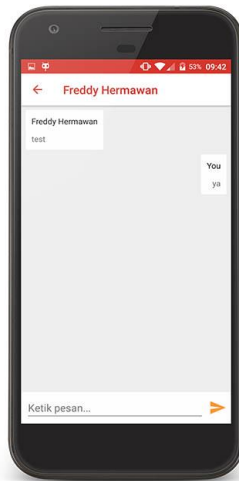
Kode Sumber 4.93 Menampilkan Muatan Halaman Ruang Chat

```

1. <LinearLayout
2.     android:orientation="vertical">
3.     <android.support.v7.widget.RecyclerView
4.         android:id="@+id/fragment_chat_message_room_layout_
recyclerview"/>
5.     <LinearLayout
6.         android:orientation="horizontal">
7.         <EditText
8.             android:id="@+id/fragment_chat_message_room_layout_
input_msg"
9.             android:hint="@string/CHATMESSAGE_HINT"/>
10.        <ImageView
11.            android:id="@+id/fragment_chat_message_room_layout_
send_msg"
12.            android:src="@drawable/ic_chat_send"/>
13.        </LinearLayout>
14.    </LinearLayout>

```

Kode Sumber 4.94 Menampilkan Halaman Ruang Chat



Gambar 4.23 Halaman Ruang Chat

4.3.23 Implementasi Fitur Daftar Tawaran

Halaman daftar tawaran digunakan oleh pengguna sebagai pelemang untuk melihat daftar tawaran yang masuk. Terdapat dua versi halaman daftar tawaran, yaitu halaman daftar tawaran saat lelang berlangsung dan halaman daftar tawaran saat sesi lelang sudah selesai.

Halaman daftar tawaran saat lelang berlangsung memungkinkan pelemang untuk memilih tawaran, membatalkan tawaran, dan melakukan blok terhadap penawar. Untuk implementasi muatan halaman daftar tawaran saat lelang berlangsung dapat dilihat pada Kode Sumber 4.96 dan implementasi antarmuka halaman daftar tawaran saat lelang berlangsung dapat dilihat pada Kode Sumber 4.97.

Sementara itu halaman daftar tawaran saat lelang selesai memungkinkan pelelang untuk memilih tawaran yang akan dijadikan sebagai pemenang lelang pada sesi tersebut. Untuk implementasi muatan halaman daftar tawaran saat lelang selesai dapat dilihat pada Kode Sumber 4.98 dan implementasi antarmuka halaman daftar tawaran saat lelang selesai dapat dilihat pada Kode Sumber 4.99.

```

1. public DaftarTawaranFragment()
2.     initializeConstants(); //inisiasi objek
3. @Override
4. public void onCreate(Bundle savedInstanceState)
5. {
6.     super.onCreate(savedInstanceState);
7.     initializeDataReceiver(); //inisiasi data receiver untuk menerima respon dari API
8.     setSocketSender(); //inisiasi socket
9. }
10. @Override
11. public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
12. {
13.     View view = inflater.inflate(R.layout.fragment_detail_barang_daftar_tawaran_layout, container, false);
14.     initializeViews(view); //memuat elemen-elemen view
15.     setSwipeRefreshLayoutProperties(); //memasang properti SwipeRefreshLayout untuk memuat API
16.     initializeTogglers(); //inisiasi objek utilitas untuk memilih, membatalkan, memblok
17.     initializeAdapter(); //inisiasi adapter
18.     setRecyclerViewProperties(); //memasang properti untuk recyclerview
19.     return view;
20. }
21. private void whenOfferListReceived()
22. {
23.     if (!listOffer.isEmpty()) {
24.         //ketika tidak ada tawaran
25.         setupViewsWhenBidExist();

```



```

26.     daftarTawaranAdapter.updateDataset(listOffer);
27.     }
28.     else {
29.         //ketika ada tawaran
30.         setupViewsWhenBidNotExist();
31.         daftarTawaranAdapter.updateDataset(listOffer);
32.     }
33. }
34. private void initializeDataReceiver()
35. {
36.     dataReceiver = new DataReceiver() {
37.         @Override
38.         public void dataReceived(Object output) {
39.             String response = output.toString();
40.             ...//menyimpan data yang didapat dari r
            espon API kedalam arraylist
41.             whenOfferListReceived();
42.         }
43.     };
44. }

```

Kode Sumber 4.95 Menampilkan Muatan Halaman Daftar Tawaran saat Lelang Berlangsung

```

1. <android.support.v4.widget.SwipeRefreshLayout
2.     android:orientation="vertical"
3.     android:id="@+id/fragment_detail_barang_daftar_tawa
       ran_layout_swipeRefreshLayout">
4.     <LinearLayout
5.         android:orientation="vertical">
6.         <android.support.v7.widget.CardView
7.             android:id="@+id/fragment_detail_barang_daftar_tawa
               ran_layout_cardview">
8.             <LinearLayout
9.                 android:orientation="vertical">
10.                 <TextView
11.                     android:text="Tawaran Tertinggi"/>
12.                 <LinearLayout
13.                     android:id="@+id/fragment_detail_barang_daftar_tawa
                           ran_layout_userproperties"
14.                     android:orientation="horizontal">
15.                 <ImageView

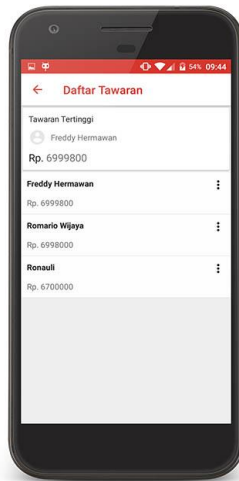
```

```

16. android:id="@+id/fragment_detail_barang_daftar_tawa
   ran_layout_avatar"/>
17. <TextView
18. android:id="@+id/fragment_detail_barang_daftar_tawa
   ran_layout_name"
19. android:text="Nama User"/>
20. </LinearLayout>
21. <LinearLayout
22. android:id="@+id/fragment_detail_barang_daftar_tawa
   ran_layout_priceproperties"
23. android:orientation="horizontal">
24. <TextView
25. android:text="Rp.  "/>
26. <TextView
27. android:id="@+id/fragment_detail_barang_daftar_tawa
   ran_layout_bidprice"/>
28. </LinearLayout>
29. <LinearLayout>
30. <TextView
31. android:id="@+id/fragment_detail_barang_daftar_tawa
   ran_layout_nobidIndicator"
32. android:text="Belum ada tawaran"/>
33. </LinearLayout>
34. </LinearLayout>
35. </android.support.v7.widget.CardView>
36. <android.support.v7.widget.RecyclerView
37. android:id="@+id/fragment_detail_barang_daftar_tawa
   ran_layout_recyclerview"
38. android:scrollbars="vertical">
39. </android.support.v7.widget.RecyclerView>
40. </LinearLayout>
41. </android.support.v4.widget.SwipeRefreshLayout>

```

Kode Sumber 4.96 Menampilkan Halaman Daftar Tawaran saat Lelang Berlangsung



Gambar 4.24 Halaman Daftar Tawaran saat Lelang Berlangsung

```

1. @Override
2. public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
3.     View view = inflater.inflate(R.layout.fragment_daftar_tawaran_final_layout, container, false);
4.     initializeConstants(); //inisiasi objek
5.     initializeFragments(); //inisiasi fragment anak

6.     getIntentData(); //mendapatkan data dari bundle intent
7.     initializeViews(view); //memuat elemen elemen view
8.     initializeDataReceivers(); //inisiasi data receiver untuk menerima respon dari API
9.     setSwipeRefreshLayoutProperties(); //memasang properti SwipeRefreshLayout untuk memuat API
10.    return view;
11. }
12. private void initializeDataReceivers() {

```

```

13.     whenOfferListLoaded = new DataReceiver() {
14.         @Override
15.         public void dataReceived(Object output) {
16.             String response = output.toString();
17.
18.             whenOfferListLoaded();
19.         }
20.     };
21. }
22. private void whenOfferListLoaded() {
23.     if (leadBidder.isWinnerStatus()) {
24.         //jika pemenang sudah terpilih, tampilkan f
ragment afterChosen
25.         fragmentManager().beginTransaction()
26.             .replace(R.id.fragment_daftar_tawar
an_final_layout, afterChosenFragment)
27.             .commit();
28.     }
29.     else {
30.         //jika pemenang belum terpilih, tampilkan f
ragment beforeChosen
31.         fragmentManager().beginTransaction()
32.             .replace(R.id.fragment_daftar_tawar
an_final_layout, beforeChosenFragment)
33.             .commit();
34.     }
35. }

```

Kode Sumber 4.97 Menampilkan Muatan Halaman Daftar Tawaran saat Lelang Selesai

```

1. <android.support.v4.widget.SwipeRefreshLayout
2.     android:id="@+id/fragment_daftar_tawaran_layout_sw
ipeRefreshLayout">
3.     <FrameLayout
4.         android:id="@+id/fragment_daftar_tawaran_final_layo
ut" />
5. </android.support.v4.widget.SwipeRefreshLayout>
6.
7. <!--fragment_daftar_tawaran_final_loaded_layout--
>
8. <LinearLayout

```

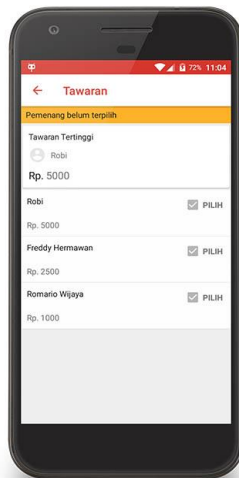
```

9.     android:orientation="vertical">
10. <LinearLayout
11.     android:id="@+id/fragment_daftar_tawaran_final_aler
        t"
12.     android:orientation="horizontal">
13. <TextView
14.     android:id="@+id/fragment_daftar_tawaran_final_aler
        t_message"/>
15. </LinearLayout>
16. <android.support.v7.widget.CardView
17.     android:id="@+id/fragment_daftar_tawaran_final_card
        view">
18. <LinearLayout
19.     android:orientation="vertical">
20. <TextView
21.     android:id="@+id/fragment_daftar_tawaran_final_sign
        "
22.     android:text="Tawaran Tertinggi"/>
23. <LinearLayout
24.     android:id="@+id/fragment_daftar_tawaran_final_user
        properties"
25.     android:orientation="horizontal">
26. <ImageView
27.     android:id="@+id/fragment_daftar_tawaran_final_avat
        ar"/>
28. <TextView
29.     android:id="@+id/fragment_daftar_tawaran_final_name
        "
30.     android:text="Nama User"/>
31. </LinearLayout>
32. <LinearLayout
33.     android:id="@+id/fragment_daftar_tawaran_final_pric
        eproperties"
34.     android:orientation="horizontal">
35. <TextView
36.     android:text="Rp. "/>
37. <TextView
38.     android:id="@+id/fragment_daftar_tawaran_final_bidp
        rice"/>
39. </LinearLayout>
40. </LinearLayout>
41. </android.support.v7.widget.CardView>
42. <android.support.v7.widget.RecyclerView

```

```
43. android:id="@+id/fragment_daftar_tawaran_final_recyclerview"  
44. android:scrollbars="vertical">  
45. </android.support.v7.widget.RecyclerView>  
46. </LinearLayout>
```

Kode Sumber 4.98 Menampilkan Halaman Daftar Tawaran saat Lelang Selesai



Gambar 4.25 Halaman Daftar Tawaran saat Lelang Selesai

4.3.24 Implementasi Fitur Daftar Blok

Halaman daftar blok digunakan untuk melihat daftar penawar yang di blok oleh pelemang pada sebuah sesi lelang. Hanya pelemang saja yang dapat melihat daftar blok. Pelemang juga dapat menonaktifkan blok yang diberikan kepada penawar. Untuk implementasi muatan halaman daftar blok dapat dilihat pada Kode Sumber 4.100 dan implementasi antarmuka halaman blok dapat dilihat pada Kode Sumber 4.101.

```

1. public DaftarBlockFragment()
2. {
3.     initializeConstants(); //inisiasi objek
4. }
5. @Override
6. public void onCreate(Bundle savedInstanceState)
7. {
8.     super.onCreate(savedInstanceState);
9.     initializeBundle(); //mendapatkan bundle dari i
    ntent
10.    initializeDataReceivers(); //inisiasi data rece
    iver
11.    initializeChildFragments(); //inisiasi fragment
    anak
12.    initializeToggler(); //inisiasi objek utilitas
    untuk membebaskan dari blok
13. }
14. @Override
15. public View onCreateView(LayoutInflater inflater, V
    iewGroup container, Bundle savedInstanceState)
16. {
17.     View view = inflater.inflate(R.layout.fragment_
    detail_barang_daftar_block_layout, container, false
    );
18.     initializeViews(view);
19.     setSwipeRefreshLayoutProperties();
20.     return view;
21. }
22. private void initializeDataReceivers()
23. {
24.     whenBlockListLoaded = new DataReceiver() {

```

```

25.         @Override
26.         public void dataReceived(Object output) {
27.             String response = output.toString();
28.             ...//menyimpan daftar blok ke dalam arr
           aylist
29.             whenBlockListLoaded();
30.         }
31.     };
32. }
33. private void whenBlockListLoaded()
34. {
35.     if (listBlock.isEmpty()) {
36.         setupFragment(emptyFragment);
37.     }
38.     else {
39.         setupFragment(noEmptyFragment);
40.     }
41. }

```

Kode Sumber 4.99 Menampilkan Muatan Halaman Daftar Blok

```

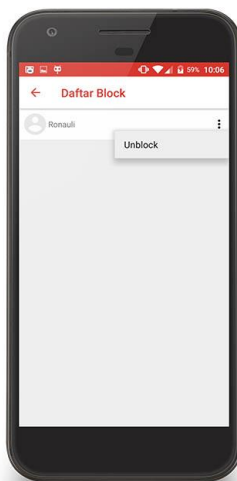
1. <android.support.v4.widget.SwipeRefreshLayout
2. android:id="@+id/fragment_detail_barang_daftar_bloc
   k_swipe_refreshLayout">
3. <FrameLayout
4. android:id="@+id/fragment_detail_barang_daftar_bloc
   k_layout">
5. </FrameLayout>
6. </android.support.v4.widget.SwipeRefreshLayout>
7.
8. <!--fragment_block_empty-->
9. <FrameLayout>
10. <LinearLayout
11. android:orientation="vertical">
12. <ImageView
13. android:src="@drawable/ic_no_block_list"/>
14. <TextView
15. android:text="Tidak ada daftar block" />
16. </LinearLayout>
17. </FrameLayout>
18.
19. <!--fragment_block_noempty-->

```



```
20. <LinearLayout
21.   android:orientation="vertical">
22.   <android.support.v7.widget.RecyclerView
23.     android:id="@+id/fragment_user_favorite_layout_recy
       clerview"
24.     android:scrollbars="vertical" />
25. </LinearLayout>
```

Kode Sumber 4.100 Menampilkan Halaman Daftar Blok



Gambar 4.26 Halaman Daftar Blok

(Halaman ini sengaja dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tentang pengujian dan evaluasi pada aplikasi yang telah dikembangkan. Pengujian yang dilakukan adalah pengujian fungsionalitas dan pengujian statistik. Pengujian fungsionalitas mengacu pada rancangan proses pada bab tiga. Pengujian statistik dilakukan untuk membuktikan bahwa penggunaan aplikasi dapat memberikan alternatif cara baru untuk berjualan dan berbelanja secara daring bagi pengguna yang menggunakan.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan dengan menggunakan perangkat bergerak sebagai berikut:

Jenis Perangkat	: Lenovo A6000
Sistem Operasi	: Android OS, v5.1.1 (Lollipop)
Ukuran Layar	: 720 x 1280 pixels, 5.0 inches
Memori	: 16 GB
RAM	: 1 GB
Chipset	: Qualcomm Snapdragon™ MSM8916
CPU	: Quad-core 1.2 GHz Cortex-A9
GPU	: Qualcomm Adreno 305

5.2 Skenario Pengujian

Subbab ini menjelaskan tentang skenario pengujian yang dilakukan. Pengujian yang dilakukan adalah pengujian fungsionalitas dan pengujian statistik. Pengujian fungsionalitas menggunakan metode kotak hitam (*black box*) yang berfokus pada hasil keluaran dari sistem berdasarkan respon masukan untuk mengetahui kesesuaian fitur dengan kebutuhan. Pengujian berbasis respon dilakukan untuk membuktikan bahwa penggunaan aplikasi dapat memberikan alternatif pilihan baru dalam berjualan

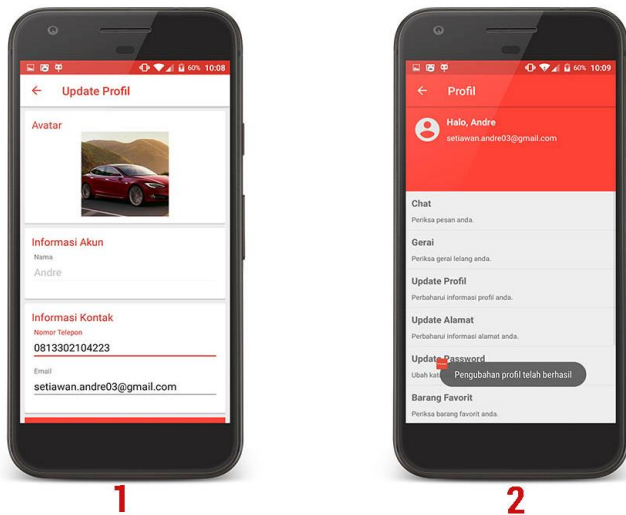
dan berbelanja daring serta nilai guna bagi pengguna yang menggunakan. Untuk mengetahui kelayakan aplikasi maka dibutuhkan kuesioner yang akan diisi oleh pengguna saat setelah mencoba menggunakan aplikasi.

5.2.1 Pengujian Fungsionalitas

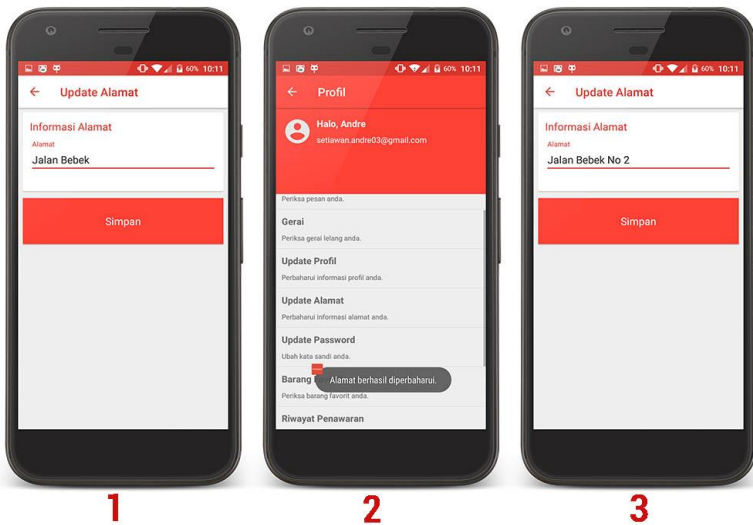
Pengujian fungsionalitas sistem dilakukan secara mandiri dengan metode *black-box*, yaitu menyiapkan beberapa skenario sebagai tolak ukur keberhasilan pengujian. Pengujian fungsionalitas yang dilakukan mengacu pada rancangan proses yang telah dijabarkan sebelumnya pada subbab 3.2.2. Berikut penjabaran dari pengujian kebutuhan fungsionalitas:

5.2.1.1 Pengujian Fungsionalitas Mengelola Data Profil Pengguna

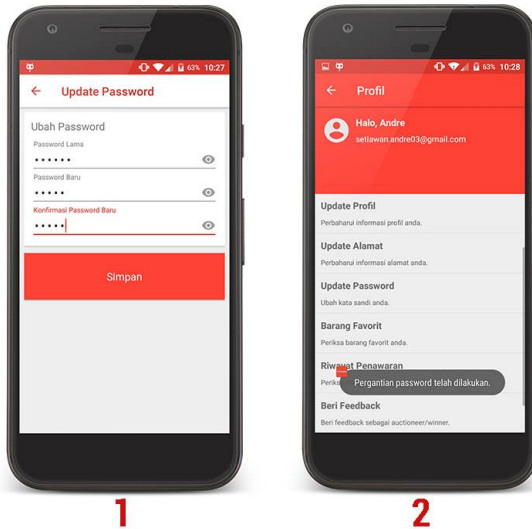
Pengujian fitur mengelola data profil pengguna dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.1 - Tabel 5.2.



Gambar 5.1 Skenario 1. Mengubah Data Profil



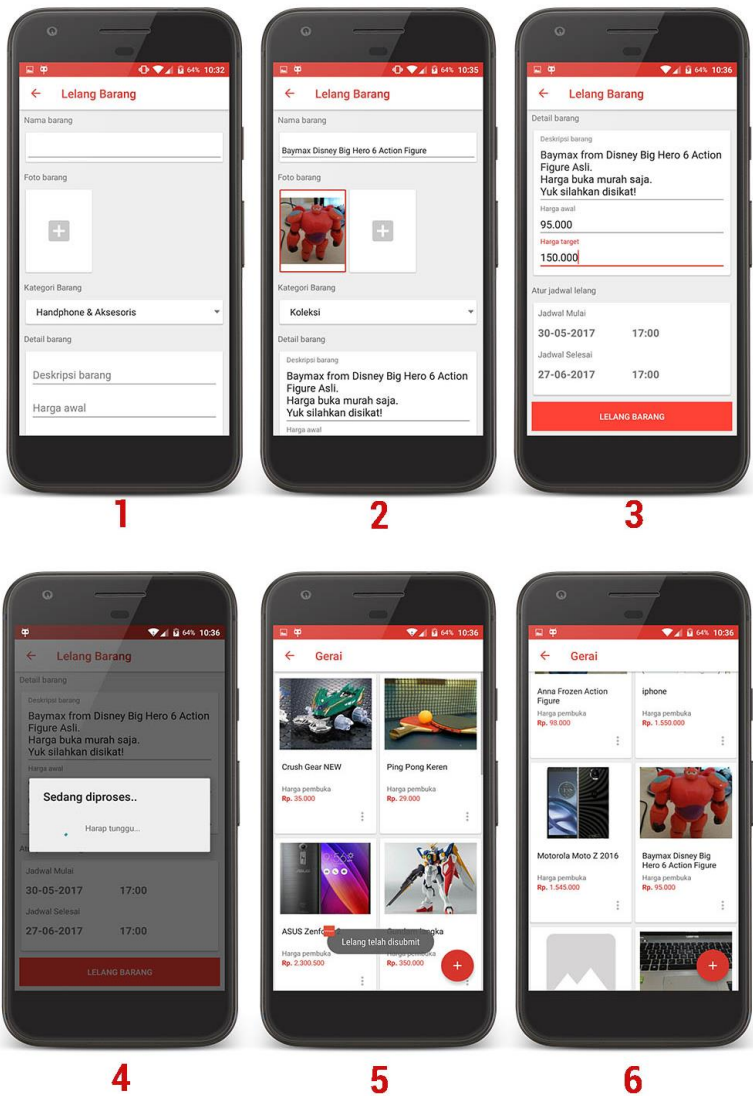
Gambar 5.2 Skenario 2. Mengubah Data Alamat



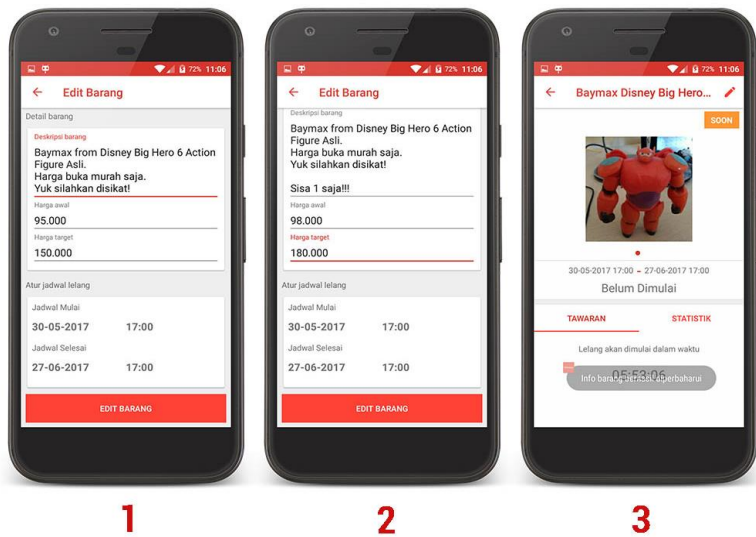
Gambar 5.3 Skenario 3. Mengubah Data Password

5.2.1.2 Pengujian Fungsionalitas Mengelola Data Barang Lelang

Pengujian fitur mengelola data barang lelang dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.3 – Tabel 5.4.



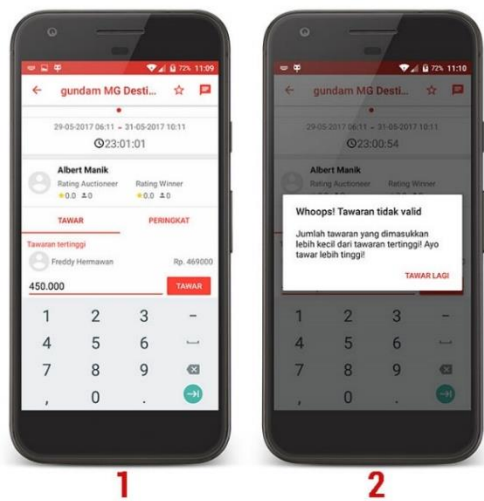
Gambar 5.4 Skenario 1. Menambah Barang Baru



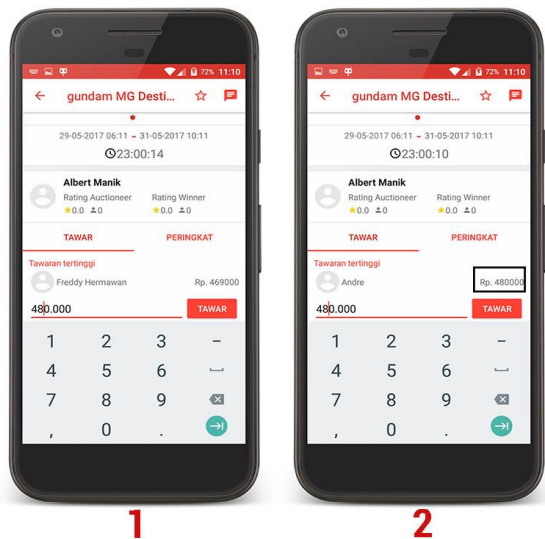
Gambar 5.5 Skenario 1. Memperbaharui Data Barang

5.2.1.3 Pengujian Fungsionalitas Mengelola Aktivitas saat Lelang Berlangsung

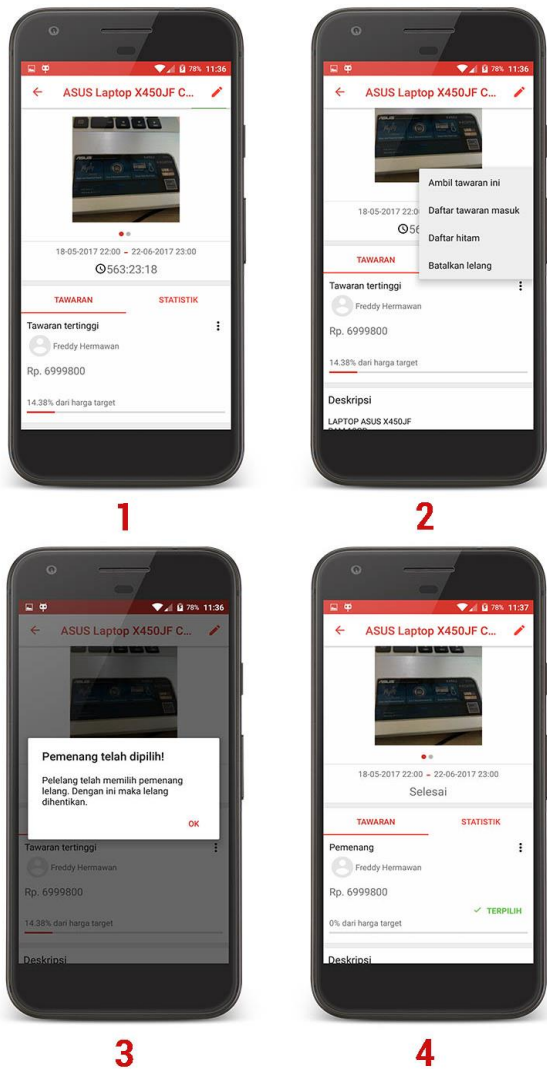
Pengujian fitur mengelola aktivitas saat lelang berlangsung dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.5 – Tabel 5.10. Skenario 1.1 – 1.2 diperuntukkan bagi pengguna sebagai penawar, sedangkan skenario 2.1 – 6 diperuntukkan bagi pengguna sebagai pelelang.



Gambar 5.6 Skenario 1.1. Memasukkan Penawaran Baru (Kondisi Ditolak)



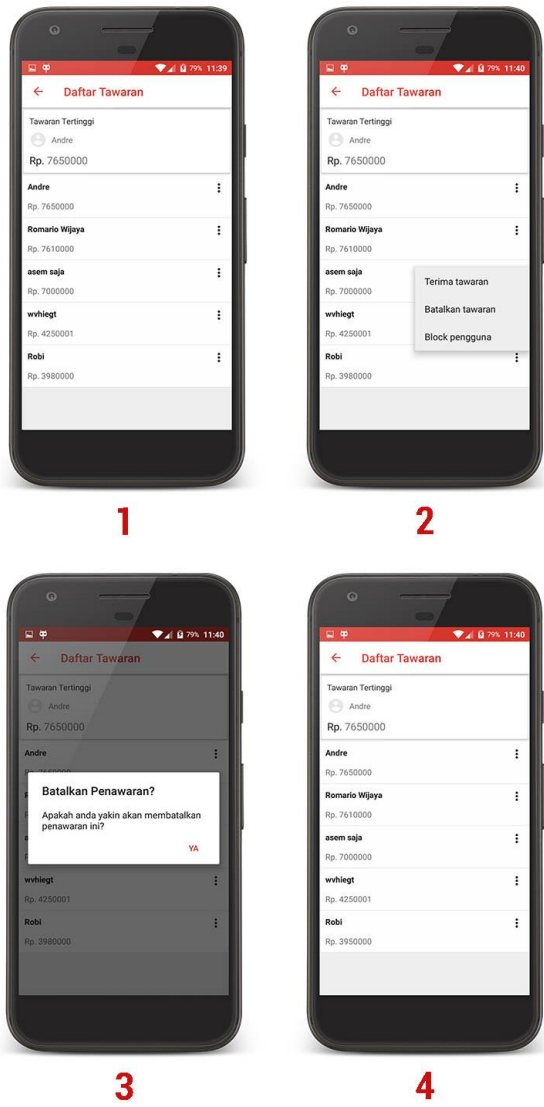
Gambar 5.7 Skenario 1.2. Memasukkan Penawaran Baru (Kondisi Diterima)



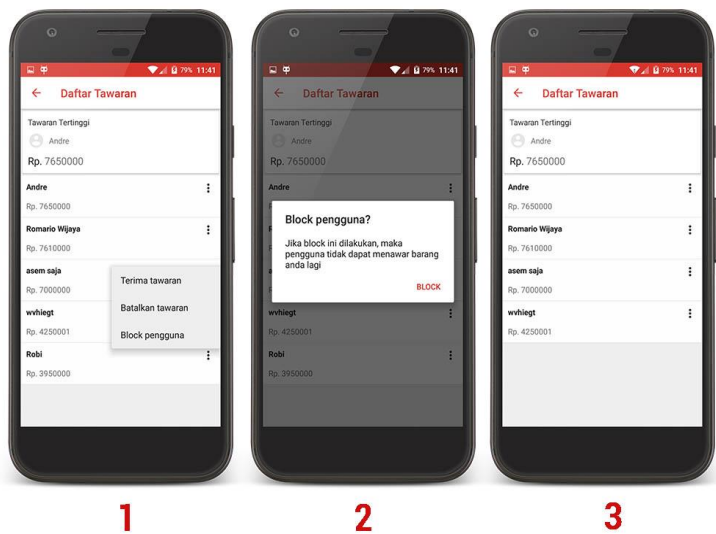
Gambar 5.8 Skenario 2.1. Memilih Tawaran Sebagai Pemenang (Melalui Halaman Detail)



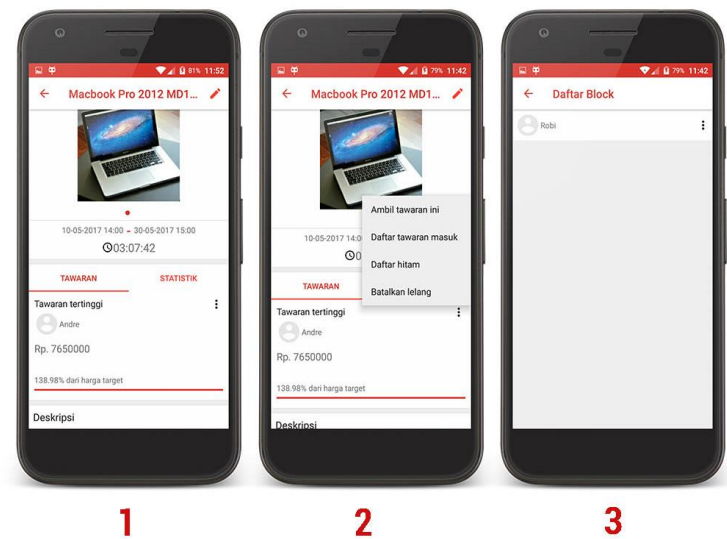
Gambar 5.9 Skenario 2.2. Memilih Tawaran Sebagai Pemenang (Melalui Halaman Daftar Tawaran)



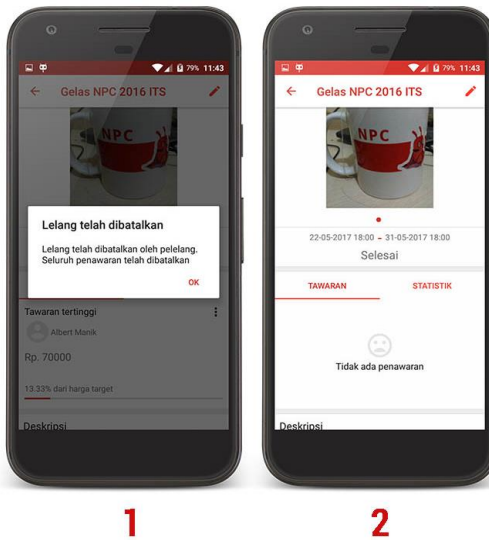
Gambar 5.10 Skenario 3. Membatalkan Tawaran dari Penawar



Gambar 5.11 Skenario 4. Melakukan Blok pada Penawar



Gambar 5.12 Skenario 5. Menampilkan Daftar Blok



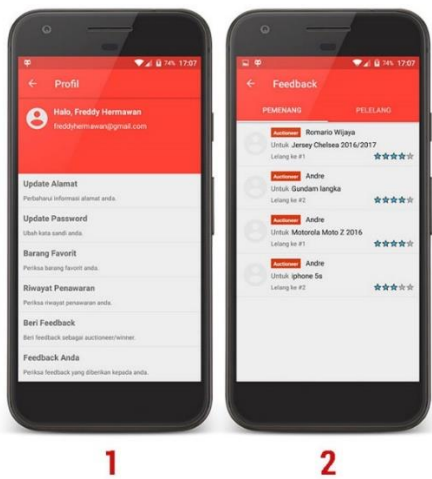
Gambar 5.13 Skenario 6. Membatalkan Lelang

5.2.1.4 Pengujian Fungsionalitas Menjalankan Aktivitas Pengguna

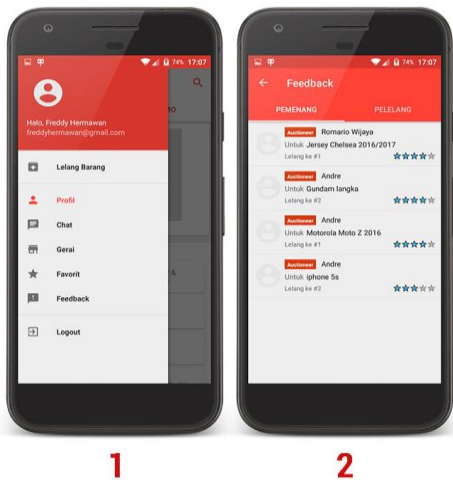
Pengujian fitur menjalankan aktivitas pengguna dibagi menjadi enam subbab, diantaranya memberikan dan menampilkan ulasan, menampilkan proses pencarian barang, menampilkan daftar dan detail riwayat penawaran, menampilkan notifikasi masuk, berkiripesan singkat, dan mengelola barang favorit.

5.2.1.4.1 Memberikan dan Menampilkan Ulasan

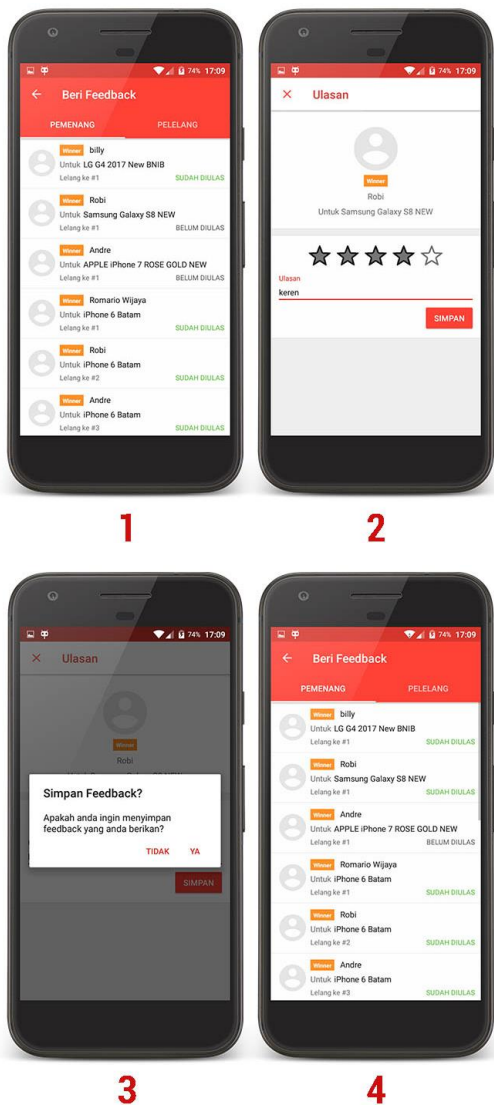
Pengujian fitur memberikan dan menampilkan ulasan dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.11 – Tabel 5.15.



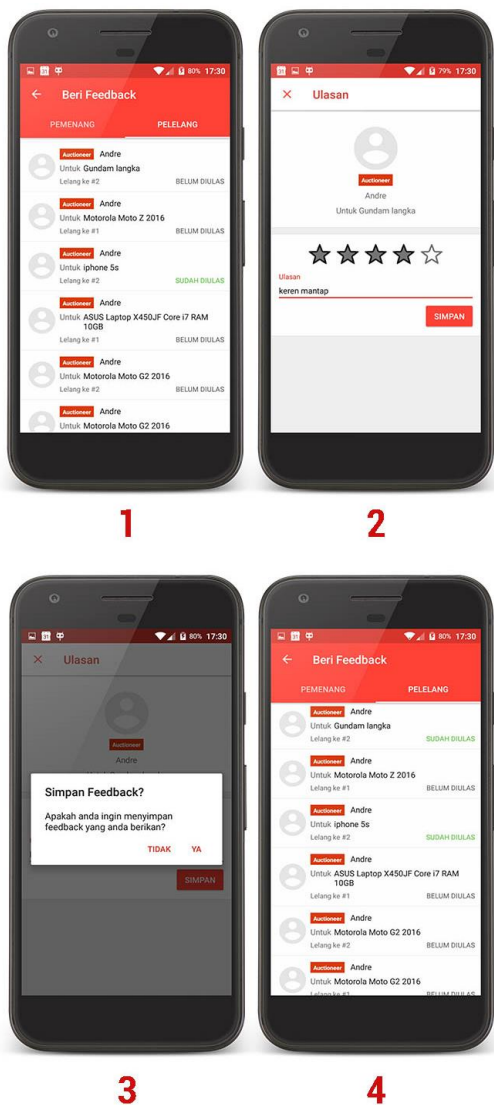
Gambar 5.14 Skenario 1. Menampilkan Ulasan (Alternatif 1)



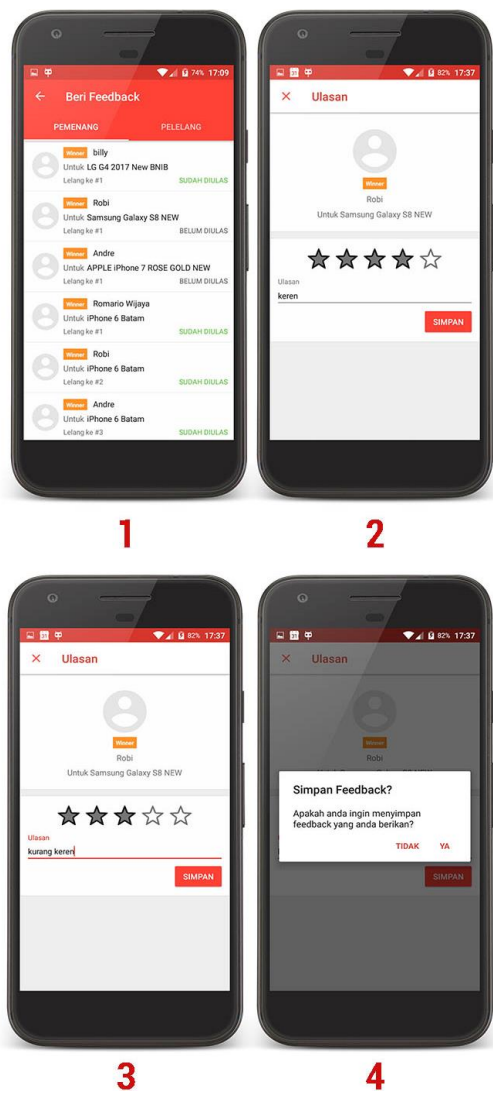
Gambar 5.15 Skenario 1. Menampilkan Ulasan (Alternatif 2)



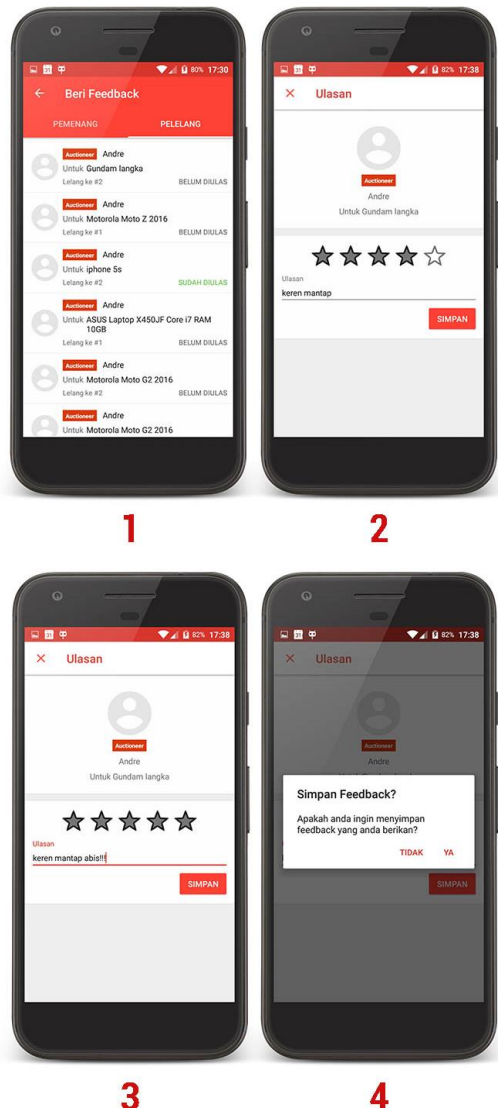
Gambar 5.16 Skenario 2. Memberikan Ulasan kepada Pemenang



Gambar 5.17 Skenario 3. Memberikan Ulasan kepada Pelelang



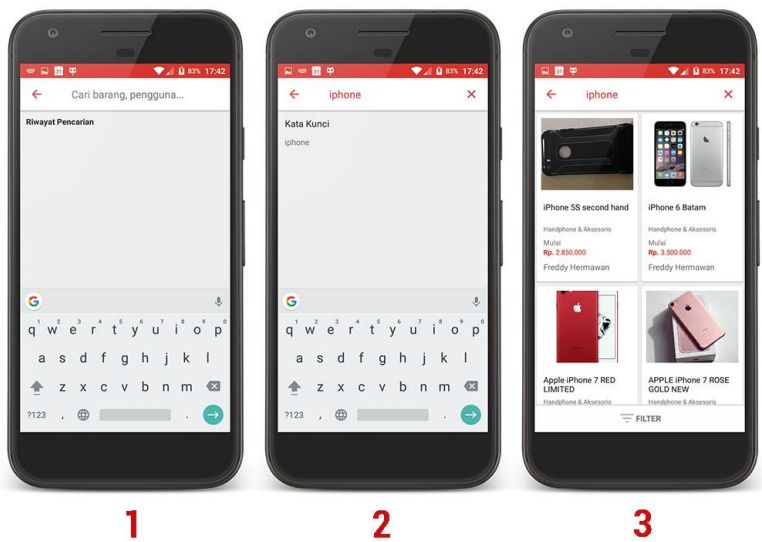
Gambar 5.18 Skenario 4. Mengubah Ulasan yang Telah Diberikan kepada Pemenang



Gambar 5.19 Skenario 5. Mengubah Ulasan yang Telah Diberikan kepada Pelelang

5.2.1.4.2 Menampilkan Proses Pencarian Barang

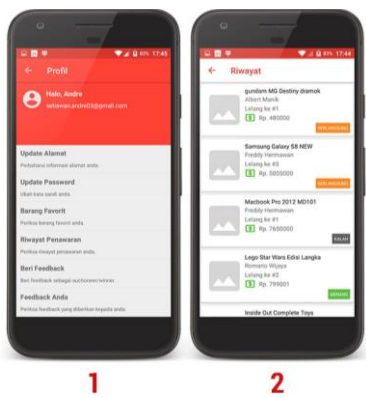
Pengujian fitur menampilkan proses pencarian barang dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.16.



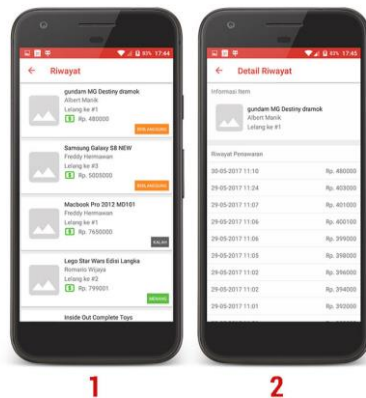
Gambar 5.20 Skenario 1. Melakukan Pencarian

5.2.1.4.3 Menampilkan Daftar dan Detail Riwayat Penawaran

Pengujian fitur menampilkan daftar dan detail riwayat penawaran dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.17 - Tabel 5.18.



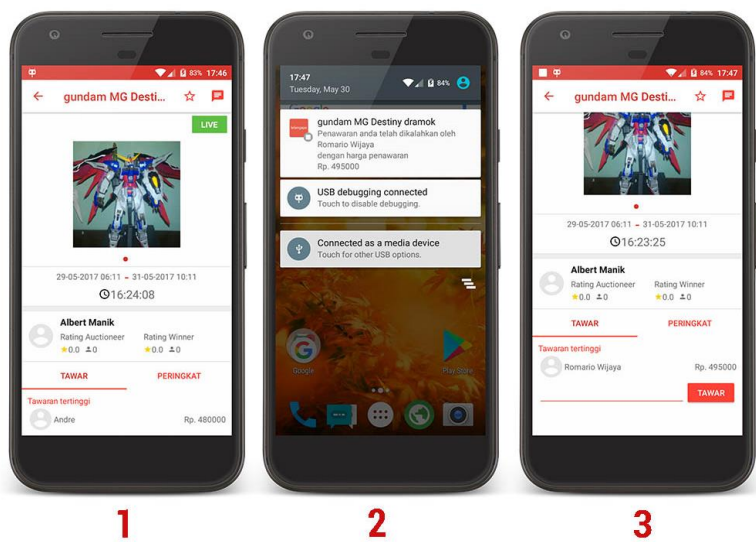
Gambar 5.21 Skenario 1. Menampilkan Daftar Riwayat



Gambar 5.22 Skenario 2. Menampilkan Detail Riwayat

5.2.1.4.4 Menampilkan Notifikasi Masuk

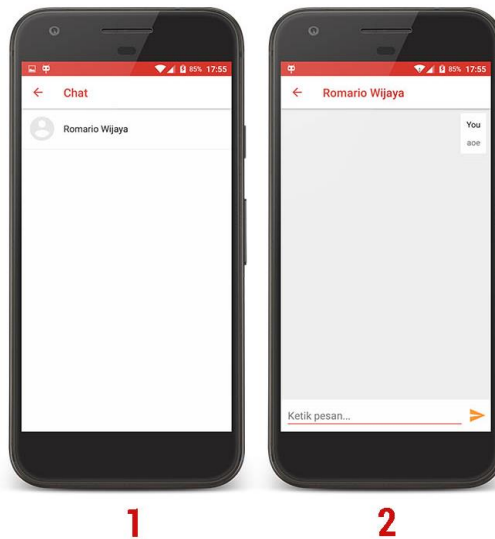
Pengujian fitur menampilkan notifikasi masuk dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.19.



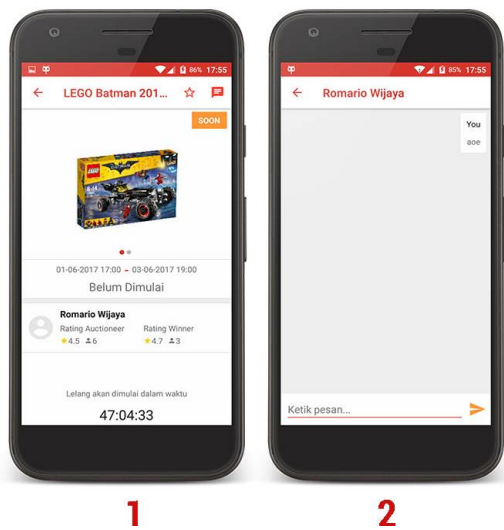
Gambar 5.23 Skenario 1. Menampilkan Notifikasi Masuk

5.2.1.4.5 Berkirim Pesan Singkat

Pengujian fitur berkirim pesan singkat dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.20.



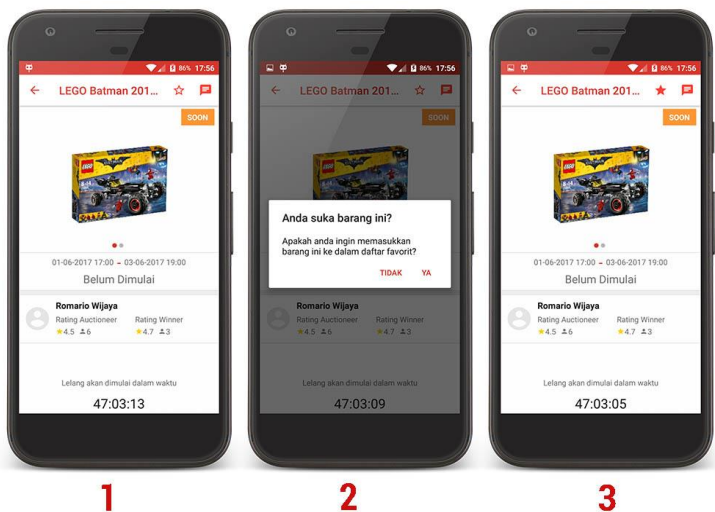
Gambar 5.24 Skenario 1. Fitur Pesan Singkat (Alernatif 1)



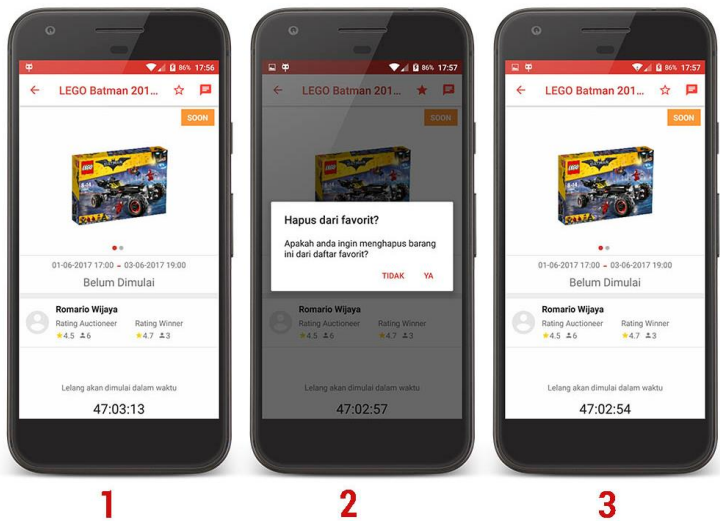
Gambar 5.25 Skenario 1. Fitur Pesan Singkat (Alernatif 2)

5.2.1.4.6 Mengelola Barang Favorit

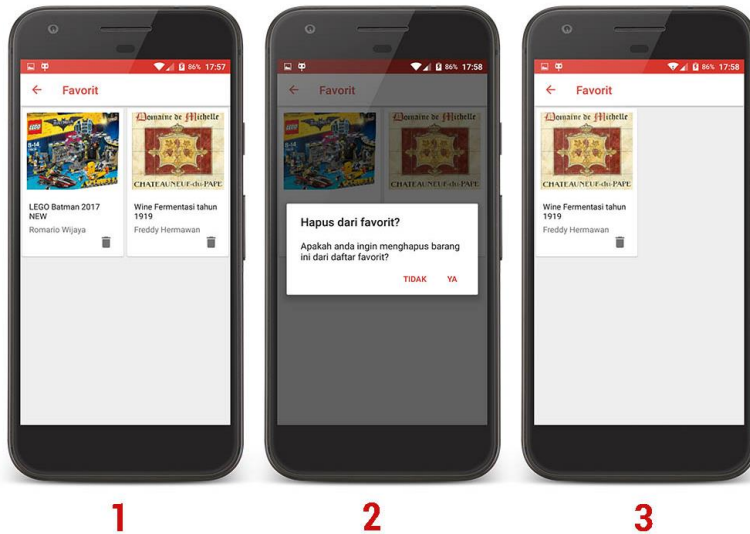
Pengujian fitur mengelola barang favorit dilakukan dengan menggunakan data uji yang disesuaikan dengan data yang diperlukan oleh pengguna. Pengujian menggunakan perangkat bergerak Lenovo A6000. Rincian pengujian fitur dicantumkan pada Tabel 5.21 – Tabel 5.22.



Gambar 5.26 Skenario 1. Memasukkan Barang Favorit



Gambar 5.27 Skenario 2. Menghapus Barang Favorit (Alternatif 1)



Gambar 5.28 Skenario 2. Menghapus Barang Favorit (Alternatif 2)

5.2.2 Pengujian Berbasis Responden

Pengujian ini bertujuan untuk membuktikan bahwa aplikasi dapat menambah alternatif baru dalam berjualan dan berbelanja daring bagi pengguna. Hal ini dapat menunjukkan jika aplikasi dapat menawarkan cara baru dalam berjualan dan berbelanja daring. Selain itu, pengujian juga bertujuan untuk menghimpun masukan yang dapat digunakan dalam pengembangan aplikasi lebih lanjut.

Proses pengujian dilakukan dengan mempublikasi perangkat lunak Lelangapa ke Google Play Store dengan status *alpha release*, dan memberikan tautan kepada lima pengguna untuk mengunduh perangkat lunak Lelangapa, mempersilahkan pengguna untuk menggunakan aplikasi, dan mengisi kuesioner mengenai kepuasan pengguna terhadap perangkat lunak, performa perangkat lunak ketika dijalankan di perangkat keras milik masing-masing pengguna, dan saran-saran yang dapat diberikan untuk pengembangan aplikasi selanjutnya.

Untuk detail mengenai responden dan kuesioner yang diisi dapat dilihat pada Lampiran A. Hasil rekap pengujian statistik dapat dilihat pada subbab 5.3.2.

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan evaluasi dari masing-masing pengujian yang telah dilakukan. Pengujian yang dilakukan adalah pengujian fungsionalitas dan pengujian statistik.

5.3.1 Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dicantumkan pada Tabel 5.23 – Tabel 5.24. Berdasarkan data pada tabel tersebut, semua skenario pengujian *blackbox* sudah berhasil dan perangkat lunak dapat berjalan dengan baik, mengirim data dan menerima respon dengan baik dan benar. Sehingga dapat disimpulkan bahwa fungsionalitas aplikasi dapat berjalan dengan baik dan benar.

5.3.2 Evaluasi Pengujian Berbasis Responden

Pengguna yang berpartisipasi dalam pengujian aplikasi telah mengirimkan kuesioner, yang terdapat pada Lampiran A. Kuesioner terdiri dari enam pertanyaan dengan penilaian skala 1 sampai 5, dan dua pertanyaan isian. Mayoritas pengguna relatif puas dengan aplikasi ini, ditunjukkan dengan banyaknya persentase pengguna yang memilih pilihan 3 (sedang) dan 4 (puas). Hal ini menunjukkan bahwa kehadiran aplikasi dapat diterima dengan baik di masyarakat, dan dapat dijadikan acuan untuk pengembangan aplikasi selanjutnya.

Pada Tabel 5.25 akan ditampilkan kuesioner penilaian aplikasi berupa skala.

**Tabel 5.1 Pengujian Fungsionalitas Fitur Mengelola Data Profil Pengguna
(1)**

ID	TA-UJ.P001
Nama Proses	Mengelola Data Profil Pengguna
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk mengelola data profil.
Skenario 1	Menguji fitur mengubah data profil
Kondisi Awal	Sistem menampilkan data profil pengguna seperti pada Gambar 5.1 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada
Langkah Pengujian	Pengguna memasukkan data yang akan diubah dan menekan tombol Ubah Profil
Hasil Yang Diharapkan	Sistem menyimpan data profil yang telah diperbaharui.
Hasil Yang Didapat	Sistem menyimpan data profil yang telah diperbaharui dan menampilkan pesan toast bahwa pembaharuan profil berhasil dilakukan.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan data profil terbaru seperti pada Gambar 5.1 bagian 2.
Skenario 2	Menguji fitur mengubah data alamat
Kondisi Awal	Sistem menampilkan data alamat pengguna seperti pada Gambar 5.2 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada.
Langkah Pengujian	Pengguna memasukkan data yang akan diubah dan menekan tombol Ubah Alamat

Tabel 5.2 Pengujian Fungsionalitas Fitur Mengelola Data Profil Pengguna (2)

ID	TA-UJ.P001
Hasil Yang Diharapkan	Sistem menyimpan data alamat yang telah diperbaharui.
Hasil Yang Didapat	Sistem menyimpan data alamat yang telah diperbaharui dan menampilkan pesan <i>toast</i> bahwa pembaharuan alamat berhasil dilakukan seperti pada Gambar 5.2 bagian 2.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan pemberitahuan seperti pada Gambar 5.2 bagian 2, dan jika pengguna membuka menu Ubah Alamat kembali, maka sistem menampilkan data alamat terbaru seperti pada Gambar 5.2 bagian 3.
Skenario 3	Menguji fitur mengubah data <i>password</i>
Kondisi Awal	Sistem menampilkan halaman ubah <i>password</i> seperti pada Gambar 5.3 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada
Langkah Pengujian	Pengguna memasukkan data <i>password</i> lama dan <i>password</i> baru.
Hasil yang Diharapkan	Sistem menyimpan data <i>password</i> terbaru.
Hasil yang Didapat	Sistem menyimpan data <i>password</i> terbaru dan menampilkan pesan <i>toast</i> bahwa <i>password</i> berhasil diperbaharui seperti pada Gambar 5.3 bagian 2.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan pemberitahuan bahwa <i>password</i> telah berhasil diubah seperti pada Gambar 5.3 bagian 2.

Tabel 5.3 Pengujian Fungsionalitas Fitur Mengelola Data Barang Lelang (1)

ID	TA-UJ.P002
Nama Proses	Mengelola Data Barang Lelang.
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk mengelola data barang lelang.
Skenario 1	Menguji fitur menambah barang baru
Kondisi Awal	Sistem menampilkan halaman formulir data barang baru seperti pada Gambar 5.4 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada.
Langkah Pengujian	Pengguna memasukkan data barang yang akan dilelang dan menekan tombol “Lelang Barang” seperti pada Gambar 5.4 bagian 2 dan 3.
Hasil Yang Diharapkan	Sistem menyimpan data barang baru yang telah dimasukkan.
Hasil Yang Didapat	Sistem menyimpan data barang baru yang telah dimasukkan dan menampilkan pesan <i>toast</i> bahwa barang sudah disimpan di server. Kemudian sistem memuat ulang halaman Gerai dan menampilkan barang yang telah dikirim seperti pada Gambar 5.4 bagian 6.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan daftar barang yang telah dimuat ulang seperti pada Gambar 5.84 bagian 6.
Skenario 2	Menguji fitur memperbaharui data barang
Kondisi Awal	Sistem menampilkan halaman formulir data barang yang akan diperbaharui seperti pada Gambar 5.5 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada.

**Tabel 5.4 Pengujian Fungsionalitas Fitur Mengelola Data Barang Lelang
(2)**

ID	TA-UJ.P002
Langkah Pengujian	Pengguna akan memasukkan pembaharuan data barang dan menekan tombol “ <i>Edit</i> Barang” seperti pada Gambar 5.5 bagian 2.
Hasil Yang Diharapkan	Sistem menyimpan data barang yang telah diperbaharui.
Hasil Yang Didapat	Sistem menyimpan data barang yang telah diperbaharui dan menampilkan pesan bahwa data barang telah diperbaharui seperti pada Gambar 5.5 bagian 3.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan pemberitahuan bahwa data barang telah diperbaharui dan kembali ke halaman detail seperti pada Gambar 5.5 bagian 3.

Tabel 5.5 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (1)

ID	TA-UJ.P003
Nama Proses	Mengelola Aktivitas saat Lelang Berlangsung.
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk mengelola aktivitas saat lelang berlangsung.
Skenario 1.1	Menguji fitur memasukkan penawaran baru (kondisi ditolak)
Kondisi Awal	Sistem menampilkan halaman detail bagi penawar seperti pada Gambar 5.6 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada.
Langkah Pengujian	Penawar akan memasukkan harga tawaran yang lebih rendah dari harga tertinggi saat ini.
Hasil Yang Diharapkan	Sistem menolak tawaran yang diajukan.
Hasil Yang Didapat	Sistem menolak tawaran yang diajukan dan menampilkan pemberitahuan seperti pada Gambar 5.6 bagian 2.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menolak tawaran yang diajukan dan menampilkan pemberitahuan seperti pada Gambar 5.6 bagian 2.
Skenario 1.2	Menguji fitur memasukkan penawaran baru (kondisi diterima)
Kondisi Awal	Sistem menampilkan halaman detail bagi penawar seperti pada Gambar 5.7 bagian 1.
Data Uji	Data uji merupakan karangan sendiri dengan mencoba aktivitas yang ada.
Langkah Pengujian	Penawar akan memasukkan harga tawaran yang lebih tinggi dari tawaran saat ini.

Tabel 5.6 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (2)

ID	TA-UJ.P003
Hasil Yang Diharapkan	Sistem menerima tawaran yang diajukan, dan mengirimkan tawaran terbaru ke semua klien yang terhubung ke di ruangan yang sama.
Hasil Yang Didapat	Sistem menerima tawaran yang diajukan, dan mengirimkan tawaran terbaru ke semua klien yang terhubung di ruangan yang sama, dan langsung menampilkan harga terbaru seperti yang terlihat pada Gambar 5.7 bagian 2.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menerima tawaran dan mengirimkan tawaran terbaru ke semua klien yang terhubung di ruangan yang sama dan mengubah harga tertinggi seperti pada Gambar 5.7 bagian 2.
Skenario 2.1	Menguji fitur memilih tawaran sebagai pemenang (melalui halaman detail)
Kondisi Awal	Sistem menampilkan halaman detail bagi pelelang seperti pada Gambar 5.8 bagian 1.
Data Uji	Data uji merupakan data tawaran tertinggi yang diberikan oleh penawar.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pelelang memilih ikon menu pop up seperti pada Gambar 5.8 bagian 2. 2. Pelelang memilih menu “Ambil Tawaran Ini”. 3. Pelelang memilih opsi OK pada dialog yang muncul setelahnya.
Hasil Yang Diharapkan	Sistem mengambil tawaran yang dipilih pengguna, dan menghentikan lelang jika lelang masih berlangsung saat pemenang dipilih.

Tabel 5.7 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (3)

ID	TA-UJ.P003
Hasil Yang Didapat	Sistem mengambil tawaran yang dipilih pengguna, menampilkan pemberitahuan bahwa pemenang telah dipilih, dan menghentikan lelang jika lelang masih berlangsung saat pemenang dipilih, seperti pada Gambar 5.8 bagian 3 dan 4.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menghentikan lelang dan menampilkan pemenang lelang seperti pada Gambar 5.8 bagian 4.
Skenario 2.2	Menguji fitur memilih tawaran sebagai pemenang (melalui halaman daftar tawaran)
Kondisi Awal	Sistem menampilkan halaman daftar tawaran seperti pada Gambar 5.9 bagian 1.
Data Uji	Data uji merupakan data daftar tawaran yang diberikan oleh penawar-penawar.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pelelang memilih ikon menu pop up pada daftar tawaran yang akan dibatalkan, seperti pada Gambar 5.9 bagian 2. 2. Pelelang memilih menu “Terima Tawaran” 3. Pelelang memilih opsi OK pada dialog yang muncul setelahnya
Hasil Yang Diharapkan	Sistem menyimpan data tawaran yang memenangkan lelang yang telah dipilih oleh pelelang.

Tabel 5.8 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (4)

ID	TA-UJ.P003
Hasil Yang Didapat	Sistem menyimpan data tawaran yang memenangkan lelang yang telah dipilih oleh pelelang, menampilkan pemberitahuan bahwa pemenang telah terpilih dan menghentikan lelang jika masih berlangsung, seperti pada Gambar 5.9 bagian 3 dan 4.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem mengembalikan ke halaman detail dan menampilkan pemenang lelang seperti pada Gambar 5.9 bagian 4.
Skenario 3	Menguji fitur membatalkan tawaran dari penawar
Kondisi Awal	Sistem menampilkan halaman daftar tawaran seperti pada Gambar 5.10 bagian 1.
Data Uji	Data uji merupakan data daftar tawaran yang diberikan oleh penawar-penawar.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pelelang memilih ikon menu pop up pada daftar tawaran yang akan dibatalkan seperti pada Gambar 5.10 bagian 2. 2. Pelelang memilih menu “Batalkan Tawaran”. 3. Pelelang memilih opsi YA pada dialog yang muncul setelahnya.
Hasil yang Diharapkan	Sistem membatalkan tawaran yang dipilih pelelang untuk dibatalkan.
Hasil yang Didapat	Sistem membatalkan tawaran yang dipilih pelelang untuk dibatalkan, memberikan pemberitahuan (Gambar 5.10 bagian 3), dan memuat ulang laman.
Hasil Pengujian	100% berhasil.

Tabel 5.9 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (5)

ID	TA-UJ.P003
Kondisi Akhir	Sistem menampilkan laman daftar tawaran terbaru, seperti pada Gambar 5.10 bagian 4.
Skenario 4	Menguji fitur melakukan blok pada penawar
Kondisi Awal	Sistem menampilkan halaman daftar tawaran (Gambar 5.11 bagian 1).
Data Uji	Data uji merupakan data daftar tawaran yang diberikan oleh penawar-penawar.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pelelang memilih ikon menu pop up pada daftar tawaran yang mana penawarnya akan di blok. 2. Pelelang memilih menu “Block Pengguna” (Gambar 5.11 bagian 2). 3. Pelelang memilih opsi BLOCK pada dialog yang muncul setelahnya.
Hasil yang Diharapkan	Sistem melakukan blok kepada penawar yang dipilih oleh pelelang dan membatalkan semua penawaran yang pernah diberikan penawar.
Hasil yang Didapat	Sistem melakukan blok kepada penawar, membatalkan semua penawaran yang pernah dilakukan penawar, dan memuat ulang laman.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan halaman daftar tawaran setelah dimuat, seperti pada Gambar 5.11 bagian 3.
Skenario 5	Menguji fitur menampilkan daftar blok
Kondisi Awal	Sistem menampilkan halaman detail bagi pelelang seperti pada Gambar 5.12 bagian 1.

Tabel 5.10 Pengujian Fungsionalitas Fitur Mengelola Aktivitas saat Lelang Berlangsung (6)

ID	TA-UJ.P003
Data Uji	Data uji merupakan data daftar penawar yang di blok oleh pelelang pada sesi lelang tersebut.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pelelang memilih ikon menu pop up seperti pada Gambar 5.12 bagian 2. 2. Pelelang memilih menu “Daftar hitam”.
Hasil yang Diharapkan	Sistem menampilkan daftar hitam pada sesi lelang tersebut.
Hasil yang Didapat	Sistem menampilkan daftar hitam pada sesi lelang tersebut.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan daftar hitam pada sesi lelang tersebut seperti pada Gambar 5.12 bagian 3.
Skenario 6	Menguji fitur membatalkan lelang
Kondisi Awal	Sistem menampilkan halaman detail bagi pelelang seperti pada Gambar 5.12 bagian 1.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pelelang memilih ikon menu pop up seperti pada Gambar 5.12 bagian 2. 2. Pelelang memilih menu “Batalkan Lelang”.
Hasil yang Diharapkan	Sistem membatalkan pengadaan lelang yang berlangsung.
Hasil yang Didapat	Sistem membatalkan pengadaan lelang yang berlangsung, dan memberikan pemberitahuan bahwa lelang sudah dibatalkan (Gambar 5.13 bagian 1), dan memuat ulang laman.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan laman yang telah dimuat ulang seperti pada Gambar 5.13 bagian 2.

Tabel 5.11 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (1)

ID	TA-UJ.P004
Nama Proses	Memberikan dan Menampilkan Ulasan.
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk memberikan dan menampilkan ulasan.
Skenario 1	Menguji fitur menampilkan ulasan sebagai pemenang dan pelemang.
Kondisi Awal	Sistem menampilkan halaman profil seperti pada Gambar 5.14 bagian 1 atau panel navigasi di halaman utama seperti pada Gambar 5.15 bagian 1.
Data Uji	Data uji merupakan daftar ulasan yang telah diberikan oleh pengguna lain.
Langkah Pengujian	Pengguna memilih menu “ <i>Feedback</i> Anda” pada halaman profil atau menu “ <i>Feedback</i> ” pada panel navigasi di halaman utama.
Hasil Yang Diharapkan	Sistem menampilkan halaman “ <i>Feedback</i> ” dan memuat daftar ulasan.
Hasil Yang Didapat	Sistem menampilkan halaman “ <i>Feedback</i> ” dan memuat daftar ulasan.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan daftar ulasan yang telah diberikan oleh pengguna lain (Gambar 5.14 bagian 2 dan 5.15 bagian 2).
Skenario 2	Menguji fitur memberikan ulasan baru kepada pemenang
Kondisi Awal	Sistem menampilkan halaman “Beri <i>Feedback</i> ” pada bagian pemenang seperti pada Gambar 5.16 bagian 1.
Data Uji	Data uji merupakan daftar ulasan yang dapat diberikan.

Tabel 5.12 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (2)

ID	TA-UJ.P004
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar ulasan yang memiliki status “BELUM DIULAS”. 2. Pengguna memasukkan ulasan berupa <i>rating</i> bintang dari 1-5 dan ulasan tertulis (Gambar 5.16 bagian 2). 3. Pengguna menekan tombol “Simpan” dan memilih opsi YA pada dialog selanjutnya.
Hasil yang Diharapkan	Sistem menyimpan ulasan yang telah diberikan.
Hasil yang Didapat	Sistem menyimpan ulasan yang telah diberikan.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem mengembalikan ke halaman “Beri <i>Feedback</i> ” dan memuat ulang laman. Ulasan yang telah diberikan sebelumnya akan diberi tanda “SUDAH DIULAS” (Gambar 5.16 bagian 4).
Skenario 3	Menguji fitur memberikan ulasan baru kepada pelemang
Kondisi Awal	Sistem menampilkan halaman “Beri <i>Feedback</i> ” pada bagian pelemang seperti pada Gambar 5.17 bagian 1.
Data Uji	Data uji merupakan daftar ulasan yang dapat diberikan.

Tabel 5.13 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (3)

ID	TA-UJ.P004
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar ulasan yang memiliki status “BELUM DIULAS”. 2. Pengguna memasukkan ulasan berupa <i>rating</i> bintang dari 1-5 dan ulasan tertulis (Gambar 5.17 bagian 2). 3. Pengguna menekan tombol “Simpan” dan memilih opsi YA pada dialog selanjutnya.
Hasil yang Diharapkan	Sistem menyimpan ulasan yang telah diberikan.
Hasil yang Didapat	Sistem menyimpan ulasan yang telah diberikan.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem mengembalikan ke halaman “Beri <i>Feedback</i> ” dan memuat ulang laman. Ulasan yang telah diberikan sebelumnya akan diberi tanda “SUDAH DIULAS” (Gambar 5.17 bagian 4).
Skenario 4	Menguji fitur mengubah ulasan yang telah diberikan kepada pemenang
Kondisi Awal	Sistem menampilkan halaman “Beri <i>Feedback</i> ” pada bagian pemenang seperti pada Gambar 5.18 bagian 1.
Data Uji	Data uji merupakan daftar ulasan yang dapat diberikan.

Tabel 5.14 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (4)

ID	TA-UJ.P004
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar ulasan yang memiliki status “SUDAH DIULAS”. 2. Pengguna memasukkan ulasan berupa <i>rating</i> bintang dari 1-5 dan ulasan tertulis. 3. Pengguna menekan tombol “Simpan” dan memilih opsi YA pada dialog selanjutnya. (Gambar 5.18 bagian 3 dan 4).
Hasil yang Diharapkan	Sistem memperbaharui ulasan.
Hasil yang Didapat	Sistem memperbaharui ulasan.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem mengembalikan ke halaman “Beri <i>Feedback</i> ” dan memuat ulang laman.
Skenario 5	Menguji fitur mengubah ulasan yang telah diberikan kepada pelemang
Kondisi Awal	Sistem menampilkan halaman “Beri <i>Feedback</i> ” pada bagian pelemang seperti pada Gambar 5.19 bagian 1.
Data Uji	Data uji merupakan daftar ulasan yang dapat diberikan.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar ulasan yang memiliki status “SUDAH DIULAS”. 2. Pengguna memasukkan ulasan berupa <i>rating</i> bintang dari 1-5 dan ulasan tertulis. 3. Pengguna menekan tombol “Simpan” (Gambar 5.19 bagian 3 dan 4).
Hasil yang Diharapkan	Sistem memperbaharui ulasan.
Hasil yang Didapat	Sistem memperbaharui ulasan.
Hasil Pengujian	100% berhasil.

Tabel 5.15 Pengujian Fungsionalitas Fitur Memberikan dan Menampilkan Ulasan (5)

ID	TA-UJ.P004
Kondisi Akhir	Sistem mengembalikan ke halaman “Beri <i>Feedback</i> ” dan memuat ulang laman.

Tabel 5.16 Pengujian Fungsionalitas Fitur Menampilkan Proses Pencarian Barang

ID	TA-UJ.P005
Nama Proses	Menampilkan Proses Pencarian Barang
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk menampilkan proses pencarian barang.
Skenario 1	Menguji fitur melakukan pencarian menggunakan kata kunci
Kondisi Awal	Sistem menampilkan halaman pencarian seperti pada Gambar 5.20 bagian 1.
Langkah Pengujian	<ol style="list-style-type: none">1. Pengguna memasukkan kata kunci pada bagian “Cari barang...”2. Pengguna menekan tombol “Enter” pada <i>keyboard</i>.
Hasil Yang Diharapkan	Sistem menampilkan hasil pencarian yang relevan dengan kata kunci.
Hasil Yang Didapat	Sistem menampilkan hasil pencarian yang relevan dengan kata kunci.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan barang-barang hasil pencarian yang relevan dengan kata kunci yang telah dikirim oleh pengguna seperti yang terlihat pada Gambar 5.20 bagian 3.

Tabel 5.17 Pengujian Fungsionalitas Fitur Menampilkan Daftar dan Detail Riwayat Penawaran (1)

ID	TA-UJ.P006
Nama Proses	Menampilkan Daftar dan Detail Riwayat Penawaran.
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk menampilkan daftar dan detail riwayat penawaran.
Skenario 1	Menguji fitur menampilkan daftar riwayat semua barang yang pernah ditawar
Kondisi Awal	Sistem menampilkan halaman profil seperti pada Gambar 5.21 bagian 1.
Langkah Pengujian	Pengguna memilih menu “Riwayat Penawaran”.
Hasil Yang Diharapkan	Sistem memuat daftar riwayat semua barang yang pernah ditawar oleh pengguna.
Hasil Yang Didapat	Sistem memuat daftar riwayat semua barang yang pernah ditawar oleh pengguna.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan daftar riwayat semua barang yang pernah ditawar oleh pengguna pada halaman Riwayat seperti pada Gambar 5.21 bagian 2.
Skenario 2	Menguji fitur menampilkan detail dari suatu riwayat
Kondisi Awal	Sistem menampilkan daftar riwayat semua barang yang pernah ditawar oleh pengguna pada halaman Riwayat seperti pada Gambar 5.22 bagian 1.
Langkah Pengujian	Pengguna memilih salah satu dari daftar riwayat semua barang yang pernah ditawar.
Hasil Yang Diharapkan	Sistem memuat detail dari riwayat yang dipilih.

Tabel 5.18 Pengujian Fungsionalitas Fitur Menampilkan Daftar dan Detail Riwayat Penawaran (2)

ID	TA-UJ.P006
Hasil Yang Didapat	Sistem memuat detail dari riwayat yang dipilih.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem menampilkan detail dari riwayat yang dipilih pada halaman Detail Riwayat seperti pada Gambar 5.22 bagian 2.

Tabel 5.19 Pengujian Fungsionalitas Fitur Menampilkan Notifikasi Masuk

ID	TA-UJ.P007
Nama Proses	Menampilkan Notifikasi Masuk
Tujuan Pengujian	Menguji fitur yang digunakan oleh sistem untuk menampilkan notifikasi masuk.
Skenario 1	Menguji fitur menampilkan notifikasi masuk saat tawaran dikalahkan
Kondisi Awal	Perangkat lunak dalam keadaan tidak sedang dibuka maupun sedang dibuka.
Langkah Pengujian	Penawar memasukkan tawaran tertinggi baru melebihi tawaran yang diajukan penawar sebelumnya, dengan catatan penawar yang mengajukan penawaran baru berbeda dengan penawar sebelumnya.
Hasil Yang Diharapkan	Sistem mengirim notifikasi kepada penawar yang tawarannya dikalahkan.
Hasil Yang Didapat	Sistem mengirim notifikasi kepada penawar yang tawarannya dikalahkan (Gambar 5.23 bagian 2).
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Notifikasi muncul pada perangkat keras milik pengguna yang tawarannya dikalahkan seperti pada Gambar 5.23 bagian 2.

Tabel 5.20 Pengujian Fungsionalitas Fitur Berkirim Pesan Singkat

ID	TA-UJ.P008
Nama Proses	Berkirim Pesan Singkat
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk berkirim pesan singkat.
Skenario 1	Menguji fitur berkirim pesan singkat
Kondisi Awal	Pengguna sedang berada dalam halaman ruang chat seperti pada Gambar 5.24 bagian 1 atau halaman detail untuk penawaran seperti pada Gambar 5.25 bagian 1.
Langkah Pengujian	<p>Jika berada di halaman ruang chat:</p> <ol style="list-style-type: none"> 1. Pengguna memilih salah satu dari daftar nama pengguna lain yang ada. 2. Pengguna mengetikkan pesan yang ingin dikirim. <p>Jika berada di halaman detail:</p> <ol style="list-style-type: none"> 1. Pengguna memilih menu yang berada di bagian pojok kanan atas layar (Gambar 5.25 bagian 1). 2. Pengguna mengetikkan pesan yang ingin dikirim.
Hasil Yang Diharapkan	Sistem mengirimkan pesan singkat kepada pengguna yang dikirim pesan.
Hasil Yang Didapat	Sistem mengirimkan pesan singkat kepada pengguna yang dikirim pesan.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Pesan terkirim dan ditampilkan di layar pada Gambar 5.24 bagian 1 atau 5.25 bagian 1.

Tabel 5.21 Pengujian Fungsionalitas Fitur Mengelola Barang Favorit (1)

ID	TA-UJ.P009
Nama Proses	Mengelola Barang Favorit
Tujuan Pengujian	Menguji fitur yang digunakan oleh pengguna untuk mengelola barang favorit.
Skenario 1	Menguji fitur memasukkan barang favorit
Kondisi Awal	Pengguna sedang berada dalam halaman detail untuk penawaran seperti pada Gambar 5.26 bagian 1.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih ikon bintang yang terdapat di pojok kanan atas layar (Gambar 5.26 bagian 1). 2. Pengguna memilih opsi YA untuk dialog yang muncul setelahnya.
Hasil Yang Diharapkan	Sistem memasukkan barang ke daftar barang favorit milik pengguna.
Hasil Yang Didapat	Sistem memasukkan barang ke daftar barang favorit milik pengguna.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Sistem memasukkan barang ke daftar barang favorit milik pengguna dan ikon bintang akan berubah (Gambar 5.26 bagian 3).
Skenario 2	Menguji fitur menghapus barang dari daftar favorit.
Kondisi Awal	Pengguna sedang berada di halaman detail untuk penawaran seperti pada Gambar 5.27 bagian 1 atau berada di halaman daftar favorit seperti pada Gambar 5.28 bagian 1.

Tabel 5.22 Pengujian Fungsionalitas Fitur Mengelola Barang Favorit (2)

ID	TA-UJ.P009
Langkah Pengujian	<p>Jika pengguna sedang berada di halaman detail:</p> <ol style="list-style-type: none"> 1. Pengguna memilih ikon bintang pada pojok kanan atas layar. 2. Pengguna memilih opsi YA untuk dialog yang muncul setelahnya. <p>Jika pengguna sedang berada di halaman daftar favorit:</p> <ol style="list-style-type: none"> 1. Pengguna memilih ikon sampah pada barang yang akan dihapus dari daftar favorit (Gambar 5.28 bagian 1). 2. Pengguna memilih opsi YA pada dialog yang muncul setelahnya.
Hasil Yang Diharapkan	Sistem menghapus barang dari favorit.
Hasil Yang Didapat	Sistem menghapus barang dari favorit.
Hasil Pengujian	100% berhasil.
Kondisi Akhir	Jika pengguna sedang berada di halaman detail, maka ikon bintang akan berubah seperti pada Gambar 5.27 bagian 3. Jika pengguna sedang berada di halaman daftar favorit, maka halaman akan dimuat ulang dan barang yang telah dihapus tidak akan nampak lagi (Gambar 5.28 bagian 3).

Tabel 5.23 Rangkuman Hasil Pengujian Fungsionalitas (1)

ID	Nama	Skenario	Hasil
TA-UJ.P001	Pengujian Fitur Mengelola Data Profil Pengguna	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
TA-UJ.P002	Pengujian Fitur Mengelola Data Barang Lelang	Skenario 1	Berhasil
		Skenario 2	Berhasil
TA-UJ.P003	Pengujian Fitur Mengelola Aktivitas saat Lelang Berlangsung	Skenario 1.1	Berhasil
		Skenario 1.2	Berhasil
		Skenario 2.1	Berhasil
		Skenario 2.2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
		Skenario 5	Berhasil
		Skenario 6	Berhasil

Tabel 5.24 Rangkuman Hasil Pengujian Fungsionalitas (2)

ID	Nama	Skenario	Hasil
TA-UJ.P004	Pengujian Fitur Memberikan dan Menampilkan Ulasan	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
		Skenario 5	Berhasil
TA-UJ.P005	Pengujian Fitur Menampilkan Proses Pencarian Barang	Skenario 1	Berhasil
TA-UJ.P006	Pengujian Fitur Menampilkan Daftar dan Detail Riwayat Penawaran	Skenario 1	Berhasil
		Skenario 2	Berhasil
TA-UJ.P007	Pengujian Fitur Menampilkan Notifikasi Masuk	Skenario 1	Berhasil
TA-UJ.P008	Pengujian Fitur Berkirim Pesan Singkat	Skenario 1	Berhasil
TA-UJ.P009	Pengujian Fitur Mengelola Barang Favorit	Skenario 1	Berhasil
		Skenario 2	Berhasil

Tabel 5.25 Rangkuman Hasil Kuisioner Kepuasan Pengguna

No.	Pertanyaan	Penilaian				
		1	2	3	4	5
1	Tampilan aplikasi secara fisik	0	0	1	3	1
2	Aplikasi memiliki tampilan yang mudah dimengerti dan dipelajari	0	0	2	2	1
3	Aplikasi mudah digunakan	0	0	3	2	0
4	Pengguna dapat mengetahui setiap fitur pada aplikasi dengan mudah	0	0	3	2	0
5	Performa keseluruhan aplikasi saat dijalankan di perangkat milik pengguna	0	1	0	1	3
6	Aplikasi dapat memberikan alternatif baru untuk jual beli barang.	0	0	1	2	2
Total		0	1	10	12	7
Persentase (%)		0	3.33	33.33	40.00	23.33

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan mengenai kesimpulan dari hasil pengujian yang telah dilakukan dan saran mengenai hal-hal yang masih bisa untuk dikembangkan dari tugas akhir ini.

6.1 Kesimpulan

Dari hasil pengamatan yang dilakukan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang sudah dilakukan, dapat disimpulkan bahwa:

1. Aplikasi Lelangapa dapat memberikan pilihan baru bagi jual beli barang dengan sistem lelang yang diusung. Aplikasi telah dibangun berdasarkan kebutuhan yang ditetapkan dan memiliki fitur inti lelang yang interaktif dan sesuai dengan lelang yang sesungguhnya, serta dapat dikembangkan lebih lanjut.
2. Arsitektur API yang telah dirancang dan dibangun dapat menunjang fungsionalitas dari aplikasi, dan semua fitur aplikasi Lelangapa yang telah dibangun dapat berjalan dengan baik pada perangkat Android.
3. Berdasarkan hasil pengujian dengan responden, aplikasi Lelangapa dapat diterima dengan baik oleh kalangan umum, dengan 63% dari jawaban kuesioner yang masuk mengatakan puas, dan 33% mengatakan cukup.
4. Aplikasi memungkinkan pengguna untuk menawar dan melelang dengan aman, dikarenakan semua *request* dan *response* antara server dan perangkat pengguna telah berada dalam jaringan aman HTTPS.
5. Aplikasi dapat tersedia untuk perangkat Android melalui Google Play Store, sehingga pengguna dengan sistem operasi Android dapat menggunakan aplikasi untuk transaksi jual beli barang.

6.2 Saran

Berikut beberapa saran dan masukan untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada perancangan, implementasi, dan pengujian yang sudah dilakukan.

1. Menyediakan fitur pembayaran dengan sistem seperti rekening bersama (*escrow*), sehingga transaksi pembayaran barang lelang dapat dilakukan di dalam aplikasi Lelangapa.
2. Memperbaiki tampilan dan alur fungsionalitas aplikasi di beberapa bagian yang dirasa masih kurang interaktif bagi pengguna.
3. Memperbaiki performa aplikasi yang dirasa masih kurang maksimal.
4. Menambah kapasitas server yang digunakan sehingga server dapat menangani lebih banyak *request* dan lebih siap untuk digunakan dalam skala yang lebih besar.

LAMPIRAN A

KUISIONER PENGGUNAAN APLIKASI

Kuisisioner penggunaan aplikasi Lelangapa pada Tabel A.1, Tabel A.2, Tabel A.3, Tabel A.4, dan Tabel A.5 diisi oleh pengguna umum yang mengikuti *alpha testing* dari Lelangapa.

Tabel A.1. Kuisisioner Penggunaan Aplikasi M. Luthfie La Roeha

No.	Pertanyaan	Penilaian				
		1	2	3	4	5
1	Tampilan aplikasi secara fisik				v	
2	Aplikasi memiliki tampilan yang mudah dimengerti dan dipelajari					v
3	Aplikasi mudah digunakan				v	
4	Pengguna dapat mengetahui setiap fitur pada aplikasi dengan mudah				v	
5	Performa keseluruhan aplikasi saat dijalankan di perangkat milik pengguna					v
6	Aplikasi dapat memberikan alternatif baru untuk jual beli barang.			v		

Tabel A.2. Kuisioner Penggunaan Aplikasi Billy

No.	Pertanyaan	Penilaian				
		1	2	3	4	5
1	Tampilan aplikasi secara fisik					v
2	Aplikasi memiliki tampilan yang mudah dimengerti dan dipelajari				v	
3	Aplikasi mudah digunakan				v	
4	Pengguna dapat mengetahui setiap fitur pada aplikasi dengan mudah			v		
5	Performa keseluruhan aplikasi saat dijalankan di perangkat milik pengguna					v
6	Aplikasi dapat memberikan alternatif baru untuk jual beli barang.					v

Tabel A.3. Kuisioner Penggunaan Aplikasi Johannes Andre

No.	Pertanyaan	Penilaian				
		1	2	3	4	5
1	Tampilan aplikasi secara fisik				v	
2	Aplikasi memiliki tampilan yang mudah dimengerti dan dipelajari			v		
3	Aplikasi mudah digunakan			v		
4	Pengguna dapat mengetahui setiap fitur pada aplikasi dengan mudah			v		
5	Performa keseluruhan aplikasi saat dijalankan di perangkat milik pengguna					v
6	Aplikasi dapat memberikan alternatif baru untuk jual beli barang.				v	

Tabel A.4. Kuisisioner Penggunaan Aplikasi Freddy Hermawan

No.	Pertanyaan	Penilaian				
		1	2	3	4	5
1	Tampilan aplikasi secara fisik			v		
2	Aplikasi memiliki tampilan yang mudah dimengerti dan dipelajari			v		
3	Aplikasi mudah digunakan			v		
4	Pengguna dapat mengetahui setiap fitur pada aplikasi dengan mudah			v		
5	Performa keseluruhan aplikasi saat dijalankan di perangkat milik pengguna		v			
6	Aplikasi dapat memberikan alternatif baru untuk jual beli barang.					v

Tabel A.5. Kuisisioner Penggunaan Aplikasi Albert Bungaran

No.	Pertanyaan	Penilaian				
		1	2	3	4	5
1	Tampilan aplikasi secara fisik				v	
2	Aplikasi memiliki tampilan yang mudah dimengerti dan dipelajari				v	
3	Aplikasi mudah digunakan			v		
4	Pengguna dapat mengetahui setiap fitur pada aplikasi dengan mudah				v	
5	Performa keseluruhan aplikasi saat dijalankan di perangkat milik pengguna				v	
6	Aplikasi dapat memberikan alternatif baru untuk jual beli barang.				v	

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] Y.-F. Kuo, S.-T. Yen, and L.-H. Chen, "Online auction service failures in Taiwan: Typologies and recovery strategies," *Electron. Commer. Res. Appl.*, vol. 10, no. 2, pp. 183–193, Mar. 2011.
- [2] J. Zhang, "The roles of players and reputation: Evidence from eBay online auctions," *Decis. Support Syst.*, vol. 42, no. 3, pp. 1800–1818, Dec. 2006.
- [3] I. Sommerville, *Software engineering. [Textbd.]*. Addison-Wesley, 1989.
- [4] N. Babich, "Mobile Design Best Practices," *UX Planet*, 14-Nov-2016. [Online]. Available: <https://uxplanet.org/mobile-design-best-practices-2d16d37ecfe>. [Accessed: 14-Jul-2017].
- [5] "99.6 percent of new smartphones run Android or iOS - The Verge." [Online]. Available: <https://www.theverge.com/2017/2/16/14634656/android-ios-market-share-blackberry-2016>. [Accessed: 14-Jul-2017].
- [6] D. Reisinger, "Apple and Samsung Stumble as Smartphone Market Soars," *Fortune*. [Online]. Available: <http://fortune.com/2017/05/23/apple-iphone-gartner-market-share/>. [Accessed: 14-Jul-2017].
- [7] "Buyer tips for auction-style listings." [Online]. Available: <http://pages.ebay.com/help/buy/tips.html>. [Accessed: 15-Jul-2017].
- [8] "Facebook Live auctions come to Singapore." [Online]. Available: <https://www.techinasia.com/singapore-facebook-live-auctions>. [Accessed: 15-Jul-2017].
- [9] "Web server performance comparison – DreamHost." [Online]. Available: <https://help.dreamhost.com/hc/en->

- us/articles/215945987-Web-server-performance-comparison. [Accessed: 15-Jul-2017].
- [10] “Bisnis Lelang (Auction) | Teori dan Perilaku Organisasi.” [Online]. Available: <http://perilakuorganisasi.com/bisnsi-lelang-auction.html>. [Accessed: 17-Jul-2017].
- [11] “Online marketplace,” *Wikipedia*. 08-Jul-2017.
- [12] “Pengenalan Javascript (dalam tutorial),” *Mozilla Developer Network*. [Online]. Available: https://developer.mozilla.org/id/docs/Web/JavaScript/Getting_Started. [Accessed: 08-Apr-2017].
- [13] “JSON.” [Online]. Available: <http://www.json.org/>. [Accessed: 08-Apr-2017].
- [14] “Android.” [Online]. Available: https://www.android.com/intl/id_id/. [Accessed: 08-Apr-2017].
- [15] “Download Android Studio and SDK Tools | Android Studio.” [Online]. Available: <https://developer.android.com/studio/index.html>. [Accessed: 08-Apr-2017].
- [16] “IBM Knowledge Center - Tables, rows, and columns.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSPK3V_6.3.0/com.ibm.swg.im.soliddb.sql.doc/doc/tables.rows.and.columns.html. [Accessed: 08-Apr-2017].
- [17] “NOSQL Databases.” [Online]. Available: <http://nosql-database.org/>. [Accessed: 08-Apr-2017].
- [18] “PostgreSQL: About.” [Online]. Available: <https://www.postgresql.org/about/>. [Accessed: 08-Apr-2017].
- [19] “Amazon Simple Storage Service (S3) — Cloud Storage — AWS,” *Amazon Web Services, Inc.* [Online].

- Available: [//aws.amazon.com/s3/](https://aws.amazon.com/s3/). [Accessed: 18-May-2017].
- [20] J. Vanderzyden, “What is Elasticsearch, and How Can I Use It?,” *Qbox.io*. [Online]. Available: <https://qbox.io/blog/what-is-elasticsearch>. [Accessed: 18-May-2017].
- [21] “Elasticsearch.” [Online]. Available: <http://www.elastic.co/products/elasticsearch>. [Accessed: 18-May-2017].
- [22] “ElasticSearch Consumers.” [Online]. Available: <http://www.elastic.co/use-cases>. [Accessed: 18-May-2017].
- [23] “What Is MongoDB?,” *MongoDB*. [Online]. Available: <https://www.mongodb.com/what-is-mongodb>. [Accessed: 08-Apr-2017].

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Andre Setiawan lahir pada 20 Juni 1995 di Kota Surabaya. Penulis menempuh pendidikan SD di SD Petra 1 (2001-2007), SMP di SMP Petra 1 (2007-2010), SMA di SMA Petra 1 (2010-2013), dan S1 di Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya (2013-2017).

Selama kuliah di Teknik Informatika ITS, penulis pernah menjadi asisten dosen pada mata kuliah Organisasi Komputer. Pada pengerjaan tugas akhir, penulis mengambil bidang Algoritma dan Pemrograman (AP). Penulis dapat dihubungi melalui alamat email **setiawan.andre03@gmail.com**.